KEY MANAGEMENT GUIDELINE

Part 1: General Guidance

Second Draft

Red text = issues to be addressed

Note to the reviewer:

The Key Management Guideline is being developed in three parts. See Section 1.5 for a description of each part. Comments may be provided at any time to GuidelineComments@nist.gov. However, NIST would like to encourage comments to be provided by August 15, 2002 on this part of the guideline.

Since this document is under development, the reader should be aware that portions of the document may change during the continuing development process.

Table of Contents

PART 1: GENERAL GUIDANCE 8

1. INTRODUCTION 8

1.1 GOAL/PURPOSE 8

1.2 AUDIENCE 9

1.3 SCOPE 9

1.4 RELATIONSHIP TO FIPS 10

1.5 CONTENT AND ORGANIZATION 11

1.6 GLOSSARY OF TERMS AND ACRONYMS 12

1.6.1 Glossary 12

1.6.2 Acronyms 21

2. KEY MANAGEMENT OVERVIEW 23

2.1 SECURITY SERVICES 23

2.1.1 Confidentiality 23

2.1.2 Data Integrity 23

2.1.3 Authentication 24

2.1.4 Authorization 24

2.1.5 Non-repudiation 24

2.1.6 Support Services 24

2.1.7 Combining Services 24

2.2. CRYPTOGRAPHIC ALGORITHMS, KEYS, AND OTHER CRYPTOGRAPHIC INFORMATION 26

2.2.1 Classes of Cryptographic Algorithms 26

2.2.2 Cryptographic Algorithm Functionality 27

2.2.2.1 Hash Function 27

2.2.2.2 Symmetric Key Algorithms used for Encryption and Decryption 27

2.2.2.2.1 Advanced Encryption Standard (AES) 28

2.2.2.2.2 Triple DES (TDES) 28

2.2.2.2.3 Modes of Operation 28

2.2.2.3 Message Authentication Codes (MACs) 28

2.2.2.3.1 MACs Using Block Cipher Algorithms 29

2.2.2.3.2 MACs Using Hash Functions 29

2.2.2.4 Digital Signature Algorithms 29

2.2.2.4.1 DSA 29

2.2.2.4.2 RSA 29
2.2.2.4.3 ECDSA 30
2.2.2.5 Key Establishment Algorithms 30
2.2.2.5.1 Discrete Log Key Agreement Schemes (Finite Field Arithmetic) 30
2.2.2.5.2 RSA Key Transport 30
2.2.2.5.3 Elliptic Curve Key Agreement and Key Transport 31
2.2.2.5.4 Key Wrapping 31
2.2.2.6 Random Number Generation 31
2.2.3 Cryptographic Information 31
2.2.3.1 Protection Requirements 31
2.2.3.1.1 Cryptographic Keys 31
2.2.3.1.2 Other Keying Material 37
2.2.3.1.3 Other Information 39
2.2.3.2 Protection Mechanisms 40
2.2.3.2.1 Protection Mechanisms for Cryptographic Information in Transit 41
2.2.3.2.1.1 Confidentiality 41
2.2.3.2.1.2 Integrity 41
2.2.3.2.1.3 Long-term Availability 42
2.2.3.2.1.4 Association with Usage or Application 42
2.2.3.2.1.5 Association with the other Entity 42
2.2.3.2.1.6 Association with Other Information 42
2.2.3.2.2 Protection Mechanisms for Information in Storage 42
2.2.3.2.2.1 Confidentiality 43
2.2.3.2.2.2 Integrity 43
2.2.3.2.2.3 Long-term Availability 44
2.2.3.2.2.4 Association with Usage or Application 44
2.2.3.2.2.5 Association with the Other Entity 44
2.2.3.2.2.6 Association with Other Information 45
2.2.3.2.3 Labeling of Cryptographic Information 45
2.2.3.2.3.1 Labels for Keys 45
2.2.3.2.3.2 Labels for Other Information 46
2.3 KEY MANAGEMENT LIFECYCLE 46
2.3.1 Pre-operational Phase 47
2.3.1.1 User Registration 47
2.3.1.2 System Initialization 47
2.3.1.3 User Initialization 47
2.3.1.4 Keying Material Installation 47

2.3.1.5 Key Establishment 48
2.3.1.5.1 Generation and Distribution of Key Pairs 48
2.3.1.5.1.1 Distribution of Static Public Keys 49
2.3.1.5.1.2 Distribution of Ephemeral Public Keys 53
2.3.1.5.1.3 Distribution of Centrally Generated Key Pairs 53
2.3.1.5.2 Generation and Distribution of Symmetric Keys 54
2.3.1.5.2.1 Key Generation 54
2.3.1.5.2.2 Key Distribution 54
2.3.1.5.2.3 Key Agreement 56
2.3.1.5.3 Generation and Distribution of Other Keying Material 56
2.3.1.5.3.1 Domain Parameters 56
2.3.1.5.3.2 Initialization Vectors 57
2.3.1.5.3.3 Shared Secrets 57
2.3.1.5.3.4 Secret and Public Seeds 57
2.3.1.5.3.5 Intermediate Results 57
2.3.1.6 Key Registration 57
2.3.2 Operational Phase 58
2.3.2.1 Normal Operational Storage 58
2.3.2.1.1 Device or Module Storage 58
2.3.2.1.2 Immediately Accessible Storage Media 58
2.3.2.2 Continuity of Operations 59
2.3.2.2.1 Backup Storage 59
2.3.2.2.2 Key Recovery 61
2.3.2.3 Key Replacement 62
2.3.2.3.1 Rekeying 62
2.3.2.3.2 Key Update 62
2.3.2.3.3 Key Derivation 62
2.3.3 Post-Operational Phase 63
2.3.3.1 Archive Storage and Key Recovery 63
2.3.3.2 User De-registration 67
2.3.3.3 Key De-registration 68
2.3.3.4 Key Destruction 68
2.3.3.5 Key Revocation 68
2.3.4 Obsolete/Destroyed Phase 69
3 GENERAL KEY MANAGEMENT GUIDANCE 70
3. 1 KEY USAGE 70

3.2 CRYPTOPERIODS 70
3.3 DOMAIN PARAMETER VALIDATION AND PUBLIC KEY VALIDATION 77
3.4 COMPROMISE OF KEYS AND OTHER KEYING MATERIAL 77
3.5 ACCOUNTABILITY 80
3.6 AUDIT 81
3.7 KEY MANAGEMENT SYSTEM SURVIVABILITY 81
3.7.1 Back-up Keys 81
3.7.2 Key Recovery 81
3.7.3 System Redundancy/Contingency Planning/Planning 82
3.7.4 Compromise Recovery 82
3.8 GUIDANCE FOR CRYPTOGRAPHIC ALGORITHM AND KEY SIZE SELECTION 82
3.8.1 Equivalent Algorithm Strengths 82
3.8.2 Defining Appropriate Algorithm Suites 84
3.8.3 Transitioning to New Algorithms and Key Sizes 87
3.9 KEY ESTABLISHMENT SCHEMES 88

APPENDIX A: CRYPTOGRAPHIC AND NON-CRYPTOGRAPHIC INTEGRITY AND AUTHENTICATION MECHANISMS 89

APPENDIX B: KEY RECOVERY 92

B.1 RECOVERY FROM STORED KEYING MATERIAL 93
B.2 RECOVERY BY RECONSTRUCTION (REDERIVATION) OF KEYING MATERIAL 93
B.3 CONDITIONS UNDER WHICH KEYING MATERIAL NEEDS TO BE RECOVERABLE 93
B.3.1 Signature Key Pair 94
B.3.1.1 Signature Verification Public Key 94
B.3.1.2 Signing Private Key 94
B.3.2 Secret Authentication Key 95
B.3.3 Authentication Key Pair 96
B.3.3.1 Authentication Public Key 96
B.3.3.2 Authentication Private Key 96
B.3.4 Data Encryption Key 96
B.3.5 Random Number Generation Key 97
B.3.6 Key Wrapping Key 97
B.3.6.1 Long-term Key Wrapping Key 97

B.3.6.2 Short-term Key Wrapping Key....97
B.3.7 Master Key used for Key Derivation and the Derived Key....98
B.3.7.1 Master Key Used for Key Derivation....98
B.3.7.2 Key Derived from a Master Key....98
B.3.8 Key Transport Key Pair....98
B.3.8.1 Key Transport Private Key....98
B.3.8.2 Key Transport Public Key....99
B.3.9 Secret Key Agreement Key....99
B.3.10 Static Key Agreement Key Pair....99
B.3.10.1 Static Key Agreement Private Key....99
B.3.10.2 Static Key Agreement Public Key....99
B.3.11 Ephemeral Key Pairs....100
B.3.11.1 Ephemeral Private Keys....100
B.3.11.2 Ephemeral Public Keys....100
B.3.12 Secret Authorization Key....100
B.3.13 Authorization Key Pair....100
B.3.13.1 Authorization Private Key....100
B.3.13.2 Authorization Public Key....100
B.3.14 Other Keying Material....101
B.3.14.1 Domain Parameters....101
B.3.14.2 Initialization Vectors (IVs)....101
B.3.14.3 Shared Secrets....101
B.3.14.4 Secret Seeds....101
B.3.14.5 Public Seeds....102
B.3.14.6 Nonces....102
B.3.14.7 Intermediate Results....102
B.3.15 Other Cryptographic Information....102
B.3.15.1 Hash Value....102
B.3.15.2 Encrypted Data....102
B.3.15.3 Message Authentication Code (MAC)....102
B.3.15.4 Digital Signature....102
B.3.15.5 Key Control Information....102
B.3.15.6 Random Number....103

B.3.15.7 Passwords 103
B.3.15.8 Audit Information 103
B.4 KEY RECOVERY SYSTEMS 103
B.5 KEY RECOVERY POLICY 104
B.6 OTHER QUESTIONS/ISSUES? 105
APPENDIX C: CRYPTOPERIODS FOR SIGNING KEY PAIRS 106
APPENDIX X: REFERENCES 107

KEY MANAGEMENT GUIDELINE

Part 1: General Guidance

1. INTRODUCTION

Cryptographic mechanisms are one of the strongest ways to provide security services for electronic applications and protocols and for data storage. The National Institute of Standards and Technology (NIST) publishes Federal Information Processing Standards (FIPS) that specify cryptographic techniques for protecting sensitive unclassified information.

Since NIST published the Data Encryption Standard (DES) in 1977, a suite of approved standardized algorithms has been growing. New classes of algorithms have been added, such as secure hash algorithms and asymmetric key algorithms for digital signatures. The suite of algorithms now provides different levels of cryptographic strength through a variety of key sizes. The algorithms may be combined in many ways to support increasingly complex protocols and applications. While a FIPS is mandatory for U.S. government agencies using cryptography for the protection of their sensitive unclassified information, a FIPS may also be followed, on a voluntary basis, by other organizations that want to implement sound security principles in their computer systems.

The proper management of cryptographic keys is essential to the effective use of cryptography for security. Keys are analogous to the combination of a safe. If the combination becomes known to an adversary, the strongest safe provides no security against penetration. Similarly, poor key management may easily compromise strong algorithms. Ultimately, the security of information protected by cryptography directly depends on the strength of the keys, the effectiveness of mechanisms and protocols associated with keys, and the protection afforded the keys. Cryptography can be rendered ineffective by the use of weak products, inappropriate algorithm pairing, poor physical security, and the use of weak protocols.

All keys need to be protected against modification, and secret and private keys need to be protected against unauthorized disclosure. Key management provides the foundation for the secure generation, storage, distribution, and destruction of keys. Another role of key management is key maintenance, specifically, the update/replacement of keys.

1.1 Goal/Purpose

Users and developers are presented with many new choices in their use of cryptographic mechanisms. Inappropriate choices may result in an illusion of security, but little or no real security for the protocol or application. Basic key management guidance is provided in [SP800-21]. This guideline expands on that guidance and provides background information and establishes frameworks to support appropriate decisions when selecting and using cryptographic mechanisms.

1.2 Audience

The audiences for the *Key Management Guideline* include system or application owners and managers, cryptographic module developers, protocol developers, and system administrators. The guideline has been provided in three parts. The different parts into which the guideline has been divided have been tailored to specific audiences.

Part 1 of the document provides general key management guidance that is intended to be useful to both system developers and system administrators. Cryptographic module developers may benefit from this general guidance through a greater understanding of key management features required to support specific intended ranges of applications. Protocol developers may identify key management characteristics associated with specific suites of algorithms and gain a greater understanding of the security services provided by those algorithms. System administrators may use this document to determine which configuration settings are most appropriate for their information.

Part 2 of the guideline is tailored for system or application owners for use in identifying appropriate organizational key management infrastructures, establishing organizational key management policies, and specifying organizational key management practices and plans.

Part 3 of the guideline is intended to provide guidance to system administrators regarding the use of cryptographic algorithms in specific applications, select products to satisfy specific operational environments, and configure the products appropriately.

Though some background information and rationale are provided for context and to support the guideline's recommendations, the guideline assumes that the reader has a basic understanding of cryptography. For background material, readers may look to a variety of NIST and commercial publications. [SP800-21] includes a brief introduction to cryptography. [SP 800-5] and [IPKI] provide an introduction to public key infrastructure. A mathematical review of cryptography and cryptographic algorithms is found in [HAC] and [AC].

1.3 Scope

This guideline encompasses cryptographic algorithms, infrastructures, protocols, and applications, and the management thereof. All cryptographic algorithms currently approved by NIST for the protection of unclassified but sensitive information are in scope, as well as algorithms actively under consideration (e.g., Diffie-Hellman). Cryptographic infrastructures that are commonly used to distribute keys for approved algorithms are considered, such as Kerberos and the X.509 public key infrastructure. The guideline addresses those protocols and applications that are widely used by Federal agencies (e.g., such as SSL).

This guideline focuses on issues involving the management of cryptographic keys: their generation, use, and eventual destruction. Related topics, such as algorithm selection and appropriate key size, cryptographic policy, and cryptographic module selection, are also included in this guideline. Some of the topics noted above are addressed in other NIST standards and guidance. For example, a public key infrastructure is addressed in [IPKI]. This guideline supplements more focused standards and guidelines.

This guideline does not address implementation details for cryptographic modules that may be used to achieve the security requirements identified. These details are addressed in [SP 800-21],

[FIPS 140-2], and [derived test requirements]. [We may need to add other areas that may be out of scope.]

This guideline often uses “requirement” terms; these terms have the following meaning in this document:

- **SHALL**: This term is used to indicate a requirement of a Federal Information Processing Standard (FIPS). Note that **SHALL** may be coupled with **NOT** to become **SHALL NOT**.
- **SHOULD**: This term is used to indicate a very important requirement. While the “requirement” is not stated in a FIPS, ignoring the requirement could result in undesirable results. Note that **SHOULD** may be coupled with **NOT** to become **SHOULD NOT**.
- **RECOMMENDED**: This term is used to indicate a recommended preference. Note that **RECOMMENDED** may be coupled with **NOT** to become **RECOMMENDED...NOT**.

1.4 Relationship to FIPS

FIPS security standards are valuable because:

a) They establish an acceptable minimal level of security for U.S. government systems. Systems that implement these standards offer a consistent level of security approved for sensitive, unclassified government data.

b) They often establish some level of interoperability between different systems that implement the standard. For example, two products that both implement the Advanced Encryption Standard (AES) cryptographic algorithm have the potential to interoperate, provided that the other functions of the product are compatible.

c) FIPS standards often provide for scalability because the U.S. government requires products and techniques that can be effectively applied in large numbers.

d) FIPS are scrutinized by the U.S. government to assure that they provide an adequate level of security. This review is performed by U.S. government experts in addition to the reviews performed by the public.

e) FIPS techniques are re-assessed every five years for their continued effectiveness. If any technique is found to be inadequate for the continued protection of government information, the standard is revised or discontinued.

f) Several of the cryptography-based FIPS (e.g., DES, TDES, SHA-1, DSA, and Cryptographic Modules) have required conformance tests. These tests are performed by accredited laboratories on vendor products that claim conformance to the standards. Vendors are permitted to modify non-conforming products so that they meet all requirements. Users of validated products can have a high degree of confidence that validated products conform to the standard.

Since 1977, NIST has built up a standards "toolbox" of FIPS security standards that form a basis for the implementation of strong cryptography. This guideline refers to many of those standards and provides guidance on how they may be properly used to protect sensitive information.

1.5 Content and Organization

The guideline is written for several different audiences and is divided into three parts.

Part 1, *General Guidance*, contains basic key management guidance. It is intended to advise developers and system administrators on the "best practices" associated with key management.

- Section 1, *Introduction*, establishes the purpose, scope and intended audience of the Key Management Guideline, and defines basic terms and acronyms, and provides background information regarding cryptographic algorithms, keying material, and the key management lifecycle. The reader should be aware that the terms used in this guideline may be defined differently in other documents.

- Section 2, *Key Management Overview*, provides an overview of key management concerns. Section 2.1, *Security Services*, defines the security services that may be provided using cryptographic mechanisms. Section 2.2, *Cryptographic Algorithms, Keys and Other Cryptographic Information*, provides essential background material. This section provides a framework for later subjects by establishing the classes of cryptographic algorithms, describing the types of functions that may be implemented using these algorithms, classifying keys and other keying material according to their function, and identifying other information that requires protection. These protection requirements are of particular interest to cryptographic module vendors and application implementers. Section 2.3, *Key Management Lifecycle*, establishes a framework for the lifecycle of a cryptographic key. This section is of particular interest to cryptographic module vendors and developers of cryptographic infrastructure services

- Section 3, *General Key Management Guidance*, discusses a variety of key management issues related to the keying material identified in Section 2. Topics discussed include key usage, cryptoperiod length, domain parameter validation and public key validation, accountability, audit, key management system survivability, guidance for cryptographic algorithm and key size selection and specific guidance for the use of the key establishment schemes identified in [FIPS XXX].

Part 2, *General Organization and Management Requirements*, is intended primarily to address the needs of system owners and managers. It provides a framework and general guidance to support establishing cryptographic key management within an organization and a basis for satisfying key management aspects of statutory and policy security planning requirements for Federal government organizations. Section 1, *Introduction*, discusses the organization of Part 2. Section 2, *Key Management Infrastructures*, describes a generic key management infrastructure. The infrastructure described is an adaptation of the Public Key Infrastructure (PKI) and other widely employed key management infrastructures to provide a Key Management Infrastructure for the management of both public and secret keying material in support of a broad range of cryptographic services. Section 3, *Key Management Policy and Practices*, provides guidance for the development of organizational key management policy statements, key management practices statements, and security plans that may be needed in support of institutional use of cryptography. Section 4, *Information Technology System Security Plans*, suggests key management planning and requirements that may be appropriate for inclusion in Major Applications Security Plans and General Support Systems Security Plans for major systems that apply cryptography. Section 5, *Key Management Plans for Cryptographic Devices or Applications*, identifies planning and

documentation that are useful for describing the set of key management products and services that may be required by a cryptographic device or application throughout its lifetime.

Part 3, *Application-Specific Key Management Guidance*, is intended to address the key management issues associated with currently available implementations.

- Section 1, *Selected Infrastructures*, is intended to provide guidance on key management issues associated with widely deployed key distribution infrastructures. This section will address key management requirements for both the infrastructure and its users. This section is of particular interest to parties developing or deploying infrastructures for key establishment. System and application owners may use this section to determine whether a particular infrastructure supports the security services required for their information.
- Section 2, *Selected Protocols*, is intended to provide guidance in the use of common cryptographic protocols. Developers may use this section to identify important implementation details. System and application owners may use this section to establish configuration parameters appropriate for their information.
- Section 3, *Selected Applications*, is intended to provide guidance on the use of cryptographic algorithms in applications. System administrators can use this text to select products and configure them appropriately.

Appendices are provided to supplement the main text in each part where a topic demands a more detailed treatment.

1.6 Glossary of Terms and Acronyms

[Note: The contents of this section are provided to assist in understanding the rest of this document.]

Definitions provided below are defined as used in this document. The same terms may be defined differently in other documents.

1.6.1 Glossary

Access authority	An entity responsible for monitoring and granting access privileges for other authorized entities.
Access control	Restricts access to resources only to privileged entities.
Accountability	A property that ensures that the actions of an entity may be traced uniquely to that entity.
Approved	FIPS-Approved and/or NIST-recommended. An algorithm or technique that is either 1) specified in a FIPS or NIST Recommendation, or 2) adopted in a FIPS or NIST Recommendation and specified either in an appendix to the FIPS or NIST Recommendation, or in a document referenced by the FIPS or NIST Recommendation.
Archive	See Key management archive.
Association	A relationship for a particular purpose. For example, a key is

	associated with the application or process for which it will be used.
Asymmetric key algorithm	See Public key cryptographic algorithm.
Attribute authority	An entity with a signing key and the authority to create attribute certificates that bind a privilege to another certificate, usually an identity certificate (i.e., a public key certificate that binds a public key with the identity of the owner of the public key).
Authentication	A process that establishes the origin of information, or determines an entity's identity.
Authentication code	A cryptographic checksum based on an Approved security function (also known as a Message Authentication Code).
Authorization	Access privileges granted to an entity; conveys an "official" sanction to perform a security function or activity.
Availability	Timely, reliable access to information by authorized entities.
Backup	A copy of information to facilitate recovery, if necessary.
Certificate	See public key certificate.
Certification authority	The entity in a Public Key Infrastructure (PKI) that is responsible for issuing certificates, and exacting compliance to a PKI policy.
Ciphertext	Data in its encrypted form.
Compromise	The unauthorized disclosure, modification, substitution or use of sensitive data (e.g., keying material and other security related information).
Confidentiality	The property that sensitive information is not disclosed to unauthorized entities.
Contingency plan	A plan that is maintained for disaster response, backup operations, and post-disaster recovery to ensure the availability of critical resources and to facilitate the continuity of operations in an emergency situation.
Cross certification	Used by one CA to certify any CA other than a CA immediately adjacent (superior or subordinate) to it in a CA hierarchy.
Cryptanalysis	1. Operations performed in defeating cryptographic protection without an initial knowledge of the key employed in providing the protection. 2. The study of mathematical techniques for attempting to defeat cryptographic techniques and information system security. This includes the process of looking for errors or weaknesses in the implementation of an algorithm or of the algorithm itself.
Cryptographic key (key)	A parameter used in conjunction with a cryptographic algorithm that determines its operation in such a way that an entity with knowledge of the key can reproduce or reverse the operation, while an entity without knowledge of the key cannot. Examples include:

- the transformation of plaintext data into ciphertext data,
- the transformation of ciphertext data into plaintext data,
- the computation of a digital signature from data,
- the verification of a digital signature,
- the computation of an authentication code from data,
- the computation of a shared secret that is used to derive keying material.

Cryptographic key component (key component)	One of at least two parameters that have the same format as a cryptographic key; parameters are combined in an Approved security function to form a plaintext cryptographic key before use.
Cryptographic module	The set of hardware, software, and/or firmware that implements Approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.
Cryptoperiod	The time span during which a specific key is authorized for use or in which the keys for a given system or application may remain in effect.
Data key, Data encrypting key	A cryptographic key that is used to cryptographically protect data (e.g., encrypt, decrypt, authenticate).
Data integrity	A property whereby data has not been altered in an unauthorized manner since it was created, transmitted or stored. In this guideline, the statement that a cryptographic algorithm "provides data integrity" means that the algorithm is used to detect unauthorized alterations.
Data origin authentication	Corroborating that the source of the data is as claimed.
Decryption	The process of changing ciphertext into plaintext using a cryptographic algorithm and key.
Digital signature	The result of a cryptographic transformation of data that, when properly implemented, provides the services of: 1. origin authentication 2. data integrity, and 3. signer non-repudiation.
Distribution	See key distribution.
Dual control	A process that uses two or more separate entities (usually persons) operating in concert to protect sensitive functions or information. No single entity is able to access or use the materials, e.g., cryptographic keys.

Encrypted key	A cryptographic key that has been encrypted using an Approved security function with a key encrypting key in order to disguise the value of the underlying plaintext key.
Encryption	The process of changing plaintext into ciphertext using a cryptographic algorithm and key.
Entity	An individual (person), organization, device or process.
Ephemeral keys	Short-lived cryptographic keys that are statistically unique to each execution of a key establishment process (e.g., unique to each message or session).
Hash-based message authentication code (HMAC)	A message authentication code that uses an Approved keyed hash function.
Hash function	A function that maps a bit string of arbitrary length to a fixed length bit string. Approved hash functions satisfy the following properties: 1. It is computationally infeasible to find any input that maps to any pre-specified output, and 2. It is computationally infeasible to find any two distinct inputs that map to the same output.
Hash value	The result of applying a hash function to information.
Initialization vector (IV)	A vector used in defining the starting point of an encryption process within a cryptographic algorithm.
Integrity	The property that sensitive data has not been modified or deleted in an unauthorized and undetected manner. In this guideline, the statement that a cryptographic algorithm "provides integrity" means that the algorithm is used to detect unauthorized modifications or deletions.
Key component	See cryptographic key component.
Key confirmation	A method of determining that an entity has the correct keying material.
Key de-registration	A stage in the lifecycle of keying material; the removal of all records of keying material that was registered by a registration authority.
Key distribution	The transport of a key and other keying material from an entity that either owns the key or generates the key to another entity that is intended to use the key.
Key encrypting key	A cryptographic key that is used for the encryption or decryption of other keys.
Key establishment	A stage in the lifecycle of keying material; the process by which cryptographic keys are securely distributed among cryptographic modules using manual transport methods (e.g., key loaders), automated methods (e.g., key transport and/or key agreement protocols), or a

	combination of automated and manual methods (consists of key transport plus key agreement).
Keying material installation	A stage in the lifecycle of keying material; the installation of keying material for operational use.
Key management	The activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs and passwords) during the entire life cycle of the keys, including their generation, storage, establishment, entry and output, and destruction.
Key management archive	A stage in the lifecycle of keying material; a repository containing keying material of historical interest.
Key Management Infrastructure	The framework and services that provide for the generation, production, distribution, control, accounting, and destruction of all cryptographic material, including symmetric keys, as well as public keys and public key certificates. It includes all elements (hardware, software, other equipment, and documentation); facilities; personnel; procedures; standards; and information products that form the system that distributes, manages, and supports the delivery of cryptographic products and services to end users.
Key Management Plan	The Key Management Plan is the document that describes for a cryptographic device or application the management of all key management products and services distributed by the Key Management Infrastructure and employed by that cryptographic device or application. The Key Management Plan documents how current and/or planned key management products and services will be supplied by the Key Management Infrastructure and used by the cryptographic application to ensure that lifecycle key management support is available.
Key Management Policy	The Key Management Policy is a high-level statement of organizational key management policies that identifies high-level structure, responsibilities, governing standards and guidelines, organizational dependencies and other relationships, and security policies.
Key management product	A key management product is a cryptographic key (symmetric or asymmetric) or certificate used for encryption, decryption, digital signature, or signature verification; and other items, such as certificate revocation lists and compromised key lists, obtained by trusted means from the same source, which validate the authenticity of keys or certificates. Software that performs either a security or cryptographic function (e.g., keying material accounting and control, random number generation, cryptographic module verification) is also considered to be a cryptographic product.
Key Management Practices Statement	The Key Management Practices Statement is a document or set of documentation that describes in detail the organizational structure,

	responsible roles, and organization rules for the functions identified in the Key Management Policy.
Key management service	A key management service is a function performed for an existing cryptographic product. Examples are key ordering, distribution, re-key, update of product attributes, and certificate revocation. Other cryptographic services include key recovery and the distribution, accounting, tracking, and control of software that performs either keying material security or cryptographic functions.
Key pair	A public key and its corresponding private key; a key pair is used with a public key algorithm.
Key recovery	A stage in the lifecycle of keying material; mechanisms and processes that allow authorized entities to retrieve keying material from key backup or archive.
Key registration	A stage in the lifecycle of keying material; the process of officially recording the keying material by a registration authority.
Key revocation	A stage in the lifecycle of keying material; a process whereby a notice is made available to affected entities that keying material **SHOULD** be removed from operational use prior to the end of the established cryptoperiod of that keying material.
Key share	[Will not be addressed at this time.]
Key Specification	A key specification documents the data format, encryption algorithms, hashing algorithms, signature algorithms, physical media, and data constraints for keys required by a cryptographic device and/or application.
Key transport	Secure transport of cryptographic keys from one cryptographic module to another module. When used in conjunction with a public key (asymmetric) algorithm, keying material is encrypted using a public key and subsequently decrypted using a private key. When used in conjunction with a symmetric algorithm, key transport is known as key wrapping.
Key update	A stage in the lifecycle of keying material; alternate storage for operational keying material during its cryptoperiod.
Key wrapping	Encrypting a symmetric key using another symmetric key (the key encrypting key). A key used for key wrapping is known as a key encrypting key.
Keying material	The data (e.g., keys and IVs) necessary to establish and maintain cryptographic keying relationships.
Label	Information that either identifies an associated parameter or provides information regarding the parameter's proper protection and use.
Manual key transport	A non-electronic means of transporting cryptographic keys by physically moving a device, document or person containing or

	possessing the key or a key component.
Message authentication code (MAC)	Data that is associated with authenticated information that allows an entity to verify the integrity of the information.
Nonce	A non-repeating value that is used only once for the same purpose.
Non-repudiation	A service that is used to provide proof of the integrity and origin of data in such a way that the integrity and origin can be verified by a third party as having originated from a specific entity in possession of the private key of the nominal originator.
Operational storage	A stage in the lifecycle of keying material; the normal storage of operational keying material during its cryptoperiod.
Operational use	A stage in the lifecycle of keying material; a stage whereby keying material is used for standard cryptographic purposes.
Password	A string of characters (letters, numbers and other symbols) that are used to authenticate an identity or to verify access authorization. Guidance on the creation and use of passwords is provided in SP-XXX.
Plaintext	Intelligible data that has meaning and can be understood without the application of decryption.
Private key	A cryptographic key, used with a public key cryptographic algorithm, that is uniquely associated with an entity and is not made public. In an asymmetric (public) cryptosystem, the private key is associated with a public key. The private key is known only by the owner of the key pair and is used to: 1. Compute the corresponding public key, 2. Compute a digital signature that may be verified by the corresponding public key, 3. Decrypt data that was encrypted by the corresponding public key, or 4. Compute a piece of common shared data, together with other information.
Proof-of-possession (POP)	A verification process whereby it is proven that the owner of a key pair actually has the private key associated with the public key.
Pseudorandom number generator (PRNG)	An algorithm that produces a sequence of bits that are uniquely determined from an initial value called a seed. The output of the PRNG "appears" to be random, i.e., the output is statistically indistinguishable from random values. A cryptographic PRNG has the additional property that the output is unpredictable, given that the seed is not known.
Public key	A cryptographic key used with a public key cryptographic algorithm that is uniquely associated with an entity and that may be made public.

	In an asymmetric (public) cryptosystem, the public key is associated with a private key. The public key may be known by anyone and is used to: 1. Verify a digital signature that is signed by the corresponding private key, 2. Encrypt data that can be decrypted by the corresponding private key, or 3. Compute a piece of shared data.
Public key certificate	A set of data that uniquely identifies an entity, contains the entity's public key and possibly other information, and is digitally signed by a trusted party, thereby binding the public key to the entity. Additional information in the certificate could specify how the key is used and its cryptoperiod.
Public key (asymmetric) cryptographic algorithm	A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that determining the private key from the public key is computationally infeasible.
Public Key Infrastructure (PKI)	A framework that is established to issue, maintain and revoke public key certificates.
Public seed	A starting value for a pseudorandom number generator. The value produced by the random number generator may be made public. The public seed is often called a “salt”.
Random Number Generator (RNG)	Produces a sequence of zero and one bits that is random in the sense, that there is no way to describe its output that is more efficient than simply listing the entire string of output. There are two basic classes: deterministic and non-deterministic. A deterministic RNG (also known as a pseudorandom number generator) consists of an algorithm that produces a sequence of bits from an initial value called a seed. A non-deterministic RNG produces output that is dependent on some unpredictable physical source that is outside human control, such as thermal noise or radioactive decay.
RECOMMENDED	This term is used to indicate a recommended preference. Note that **RECOMMENDED** may be coupled with **NOT** to become **RECOMMENDED...NOT**.
RSA	The public key algorithm that was developed by Rivest, Shamir, and Adleman.
Salt	A value that may be publicly known and is sometimes used in cryptographic processes.
Secret key	A cryptographic key that is used with a secret key (symmetric) cryptographic algorithm, that is uniquely associated with one or more entities and **SHOULD NOT** be made public. The use of the term

	"secret" in this context does not imply a classification level, but rather implies the need to protect the key from disclosure.
Secret seed	A secret value that used to initialize a pseudorandom number generator. The resulting value from the random number generator **SHOULD** remain secret or private.
Secure communication protocol	A communication protocol that provides the appropriate confidentiality, authentication and content integrity protection.
Security domain	A system or subsystem that is under the authority of a single trusted authority. Security domains may be organized (e.g., hierarchically) to form larger domains.
Security policy	Defines the threats that a system **SHOULD** address and provides high level mechanisms for addressing those threats.
Security services	Mechanisms used to provide confidentiality, data integrity, authentication or non-repudiation of information.
SHALL	This term is used to indicate a requirement of a Federal Information processing Standard (FIPS). Note that **SHALL** may be coupled with **NOT** to become **SHALL NOT**.
Shared secret	A value that is generated during a key agreement process; the shared secret is typically used to derive keying material for a symmetric key algorithm.
SHOULD	This term is used to indicate a very important requirement. While the "requirement" is not stated in a FIPS, ignoring the requirement could result in undesirable results. Note that **SHOULD** may be coupled with **NOT** to become **SHOULD NOT**.
Signature generation	Uses a digital signature algorithm and a private key to generate a digital signature on data.
Signature verification	Uses a digital signature algorithm and a public key to verify a digital signature.
Split knowledge	A procedure whereby a cryptographic key is handled as multiple key components from the time that the key or the separate key components are generated until the key components are combined for use. Each key component provides no knowledge of the ultimate key. The key may be created and then split into the key components, or may be created as separate key components. The key components are output from the generating cryptographic module(s) to separate entities for individual handling, and subsequently input separately into the intended cryptographic module and combined to form the ultimate key. Note: A suitable combination function is not provided by simple concatenation; e.g., it is not acceptable to form an 80 bit key by concatenating two 40-bit key components.
Static keys	Static keys are relatively long-lived and are common to a number of

	executions of a given algorithm.
Statistically unique	For the generation of *n*-bit quantities, the probability of two values repeating is less than or equal to the probability of two *n*-bit random quantities repeating. Thus, an element chosen from a finite set of 2^n elements is said to be statistically unique if the process that governs the selection of this element provides a guarantee that for any integer $L \leq 2^n$ the probability that all of the first *L* selected elements are different is no smaller than the probability of this happening when the elements are drawn uniformly at random from the set.
Symmetric key	A single cryptographic key that is used with a secret (symmetric) key algorithm.
Symmetric key algorithm	See Secret key cryptographic algorithm.
System initialization	A stage in the lifecycle of keying material; setting up and configuring a system for secure operation.
Threat	Any circumstance or event with the potential to adversely impact a system through unauthorized access, destruction, disclosure, modification of data or denial of service.
Unauthorized disclosure	An event involving the exposure of information to entities not authorized access to the information.
User initialization	A stage in the lifecycle of keying material; the process whereby a user initializes its cryptographic application (e.g., installing and initializing software and hardware).
User registration	A stage in the lifecycle of keying material; a process whereby an entity becomes a member of a security domain.
X.509 certificate	The ISO/ITU-T X.509 standard defined two types of certificates – the X.509 public key certificate, and the X.509 attribute certificate. Most commonly (including this document), an X.509 certificate refers to the X.509 public key certificate.
X.509 public key certificate	The public keys for a user (or device) and a name for the user (or device), together with some other information, rendered unforgeable by the digital signature of the certification authority that issued the certificate, encoded in the format defined in the ISO/ITU-T X.509 standard.

1.6.2 Acronyms

The following abbreviations and acronyms are used in this standard:

AES	Advanced Encryption Standard specified in FIPS 197.
ANSI	American National Standards Institute
CA	Certification Authority

CIO	Chief Information Officer
CKL	Compromised Key List
CN	Client Node
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CSN	Central Service Node
DEK	Data Encryption Key
DSA	Digital Signature Algorithm specified in FIPS 186-2.
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard.
HMAC	Keyed-Hash Message Authentication Code specified in FIPS 198.
IV	Initialization Vector.
KEK	Key Encrypting Key
KMI	Key Management Infrastructure
KMP	Key Management Policy
KMPS	Key Management Practices Statement
LAN	Local Area Network
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
PKI	Public Key Infrastructure
POP	Proof-of-Possession
PRNG	Pseudorandom Number Generator
PSN	Production Service Node
PSRN	Primary Services Node
RA	Registration Authority
RNG	Random Number Generator
TDES	Triple Data Encryption Standard; Triple DES

2. KEY MANAGEMENT OVERVIEW

This key management overview is organized so that each section provides information that will enable an understanding of subsequent sections.

Section 2.1, *Security Services*, defines the security services that may be provided using cryptographic mechanisms.

Section 2.2, *Cryptographic Algorithms, Keys and Other Cryptographic Information,* provides essential background material. This section provides a framework for later subjects by establishing classes of cryptographic algorithms, describing the types of functions that may be implemented using these classes, and classifying keys according to their function. As part of the classification of keys, protection requirements are identified for different types of keys. These protection requirements may be of particular interest to cryptographic module vendors and application implementers.

Section 2.3, *Key Management Lifecycle*, establishes a framework for the lifecycle of a cryptographic key. This section may be of particular interest to cryptographic module vendors and developers of cryptographic infrastructure services (e.g., Kerberos or public key infrastructure).

2.1 Security Services

Cryptography may be used to perform several basic security services: confidentiality, data integrity, authentication, authorization and non-repudiation. These services may also be required to protect cryptographic keying material. In addition, there are other cryptographic and non-cryptographic mechanisms that are used to support these security services. In general, a single cryptographic process often provides more than one service (e.g., the use of digital signatures can provide integrity, authentication and non-repudiation).

Appendix A.1 provides a discussion of cryptographic and non-cryptographic methods for providing data integrity and authentication services in communication protocols.

2.1.1 Confidentiality

Confidentiality is the property whereby information is not disclosed to unauthorized parties. Secrecy and privacy are terms that are often used synonymously with confidentiality.

Confidentiality that is achieved using cryptography makes use of encryption to render the information unintelligible except by authorized entities. The information may become intelligible again by using decryption.

2.1.2 Data Integrity

Data integrity is the property whereby data is not modified in an unauthorized manner; this includes the insertion, deletion and substitution of data. Cryptographic algorithms can be used to detect errors in the information (i.e., detect that the information has been modified). Absolute protection against modification is not possible. However, well-designed detection mechanisms can be used to support the maintenance of data integrity.

Cryptographic mechanisms, such as message authentication codes or digital signatures, can be used to detect (with a high probability) both accidental modifications (e.g., modifications that sometimes occur during noisy transmissions or by hardware memory failures), and deliberate modifications by an adversary with a very high probability. Non-cryptographic mechanisms are also often used to detect accidental modifications, but cannot be relied upon to detect deliberate modifications. A more detailed treatment of this subject is provided in Appendix A.1.

In this guideline, the statement that a cryptographic algorithm "provides data integrity" means that the algorithm is used to detect unauthorized alterations.

2.1.3 Authentication

Authentication is a service that is used to establish the origin of information. That is, authentication services verify the identity of the user or system that created information (e.g., a transaction or message). This service supports the receiver in security relevant decisions, such as "Is the sender an authorized user of this system?" or "Is the sender permitted to read sensitive information?" Several cryptographic mechanisms may be used to provide authentication services. Most commonly, authentication is provided by digital signatures or message authentication codes; some key agreement techniques also provide authentication.

2.1.4 Authorization

Authorization is concerned with providing an official sanction or permission to perform a security function or activity. Normally, authorization is granted following a process of authentication. A non-cryptographic analog of the interaction between authentication and authorization is the examination of an individual's credentials to establish their identity (authentication); upon proving identity, the individual is then provided with the key or password that will allow access to some resource, such as a locked room (authorization).

2.1.5 Non-repudiation

Non-repudiation is a service that is used to provide proof of the integrity and origin of data in such a way that the integrity and origin can be verified by a third party. This service prevents an entity from successfully denying involvement in a previous action. Non-repudiation is provided cryptographically by the use of a digital signature that is calculated by a private key known only by the entity that computes the digital signature.

2.1.6 Support Services

Cryptographic security services often require supporting services. For example, cryptographic services often require the use of key establishment and random number generation services.

2.1.7 Combining Services

In many applications a combination of cryptographic services (confidentiality, data integrity, authentication, authorization, and non-repudiation) is desired. Confidentiality and authentication are frequently combined so that the receiver of an encrypted message knows that the information was protected from disclosure and that it came from a particular entity. Similarly, data integrity and authentication are often combined. A service may also include another service. For example, non-repudiation by the originator of data requires that the originator be correctly identified (i.e., authenticated) and that the data have integrity.

Designers of secure systems often begin by considering which security services are needed to protect the information contained within and processed by the system. After these services have been determined, the designer then considers what mechanisms will best provide these services. Not all mechanisms are cryptographic in nature. For example, physical security may be used to protect the confidentiality of certain types of data, and identification badges or biometric identification devices may be used for entity authentication. However, cryptographic mechanisms consisting of algorithms, keys, and other keying material often provide the most cost-effective means of protecting the security of information. This is particularly true in applications where the information would otherwise be exposed to unauthorized entities.

When properly implemented, some cryptographic algorithms provide multiple services. The following examples illustrate this case:

1. The use of a symmetric encryption algorithm (Section 2.2.2.2) may provide authentication as well as confidentiality if the secret keys are unique to each pair of users.
2. A message authentication code (Section 2.2.2.3) may provide authentication as well as data integrity if the secret keys are unique to each pair of users.
3. A digital signature algorithm (Section 2.2.2.4) can provide authentication and data integrity as well as non-repudiation.
4. A digital signature on a public key certificate can be used to provide entity authentication and support confidentiality via key establishment.
5. A digital signature on a public key certificate can be used to provide entity authentication and integrity protection.

As a more concrete example of number 4 above, suppose that the secure exchange of information between pairs of Internet entities is needed. Some of the exchanged information requires just integrity protection, while other information requires both integrity and confidentiality protection. It is also a requirement that each entity that participates in an information exchange know the identity of the other entity.

The designers of the system decided that a Public Key Infrastructure (PKI) would be established and that each entity wishing to communicate securely would be required to physically authenticate their identity at a Registration Authority (RA). This authentication would require the presentation of proper credentials such as a driver's license, passport, or birth certificate. The authenticated individuals would then generate a static key pair in a smart card. The static public key of each net member would be transferred from the smart card to the RA where it is incorporated with the user identity and other information into a digitally signed message for transmission to a Certificate Authority (CA). The CA would then compose the user's public key certificate by signing the public key of the user and the user's identity along with other information. This certificate is returned to the public key owner so that it may be used in conjunction with the private key (under the sole control of the owner) for entity authentication and key establishment purposes.

Any two entities wishing to communicate may exchange public key certificates that are checked by verifying the CA signature on the certificate (using the CA public key). The static public key of each entity and each entity's own private key is then used in a key establishment scheme to produce a secret value shared between the two entities. The shared secret may then be used to derive one or more shared symmetric keys. One of the shared symmetric keys could be used to

encrypt all information that requires confidentially protection. The receiver of such protected data would have assurance that the data came from the other entity indicated by the public key certificate (entity authentication) and that the data was protected from unauthorized disclosure (confidentiality protection).

As a more concrete example of number 5 above, a static key pair and corresponding certificate could be established for each entity to exchange data that requires only integrity protection and entity authentication. The private key of the sender would be used to sign the data, and the sender's public key would be used by the receiver to verify the signature.

These examples gives a basic sketch of how cryptographic algorithms may be used to provide multiple security services. However, it can be easily seen that the security of this system depends on many factors including:

a. The degree to which a driver's license, passport, or birth certificate correctly authenticates an individual,

b. The strength of the cryptographic algorithms used,

c. The degree of trust placed in the RA and the CA, and

d. The strength of the key establishment protocols.

e. The care of the users to protect their smart cards (and thus, their keys) from unauthorized use.

2.2. Cryptographic Algorithms, Keys, and Other Cryptographic Information

FIPS-approved or NIST-recommended cryptographic algorithms **SHOULD** be used whenever cryptographic services are required. These Approved algorithms have received an intensive security analysis prior to their approval and continue to be examined to determine that the algorithms provide adequate security. Most cryptographic algorithms require cryptographic keys or other keying material. In some cases, an algorithm may be strengthened by the use of larger keys. This guideline advises the users of cryptographic mechanisms on the approved choices of algorithms and key sizes.

This section describes the Approved cryptographic algorithms that provide security services such as confidentiality, data integrity, authentication, authorization, non-repudiation, and algorithms that support these services. A general discussion on keys, other keying material and other related information requiring protection follows the overview of cryptographic algorithms.

2.2.1 Classes of Cryptographic Algorithms

There are three basic classes of Approved cryptographic algorithms, hash algorithms, symmetric key algorithms and asymmetric key algorithms. The classes are defined by the number of cryptographic keys that are used in conjunction with the algorithm.

Cryptographic hash algorithms (i.e., hash functions) require no keys. Hash algorithms generate a small message digest (hash value) from a (possibly) large message and are used as components in many cryptographic processes, such as Message Authentication Codes (MACs) (Section 2.2.2.3), digital signatures (Section 2.2.2.4), key establishment (Section 2.2.2.5), and random number generation (Section 2.2.2.6)).

Symmetric key algorithms (sometimes known as secret key algorithms) use a single key to transform data. Symmetric keys are often known by more than one entity; however, the key **SHOULD NOT** be known by entities that are not authorized access to the data protected by that algorithm and key. Symmetric key encryption algorithms are used:

- To provide data confidentiality (Section 2.2.2.2) - the same key is used to encrypt and decrypt data.
- To provide authentication and integrity services (Section 2.2.2.3) in the form of Message Authentication Codes (MACs) - the same key is used to generate the MAC and to validate it. MACs, normally employ either a symmetric key encryption algorithm or a cryptographic hash function as their cryptographic primitive.
- As part of the key establishment process (Section 2.2.2.5).
- To generate pseudorandom numbers (Section 2.2.2.6).

Asymmetric key algorithms, commonly known as public key algorithms, use two related keys (i.e., a key pair) to perform their functions: a public key and a private key. The public key may be known by anyone; the private key **SHOULD** be under the sole control of the entity that "owns" the key pair. Even though the public and private keys of a key pair are related, knowledge of the public key does not reveal the private key. Public key algorithms are commonly used in the computation of digital signatures (Section 2.2.2.4) and in the establishment of cryptographic keying material (Section 2.2.2.5).

2.2.2 Cryptographic Algorithm Functionality

Security services are fulfilled using a number of different algorithms. In many cases, the same algorithm may be used to provide multiple services.

2.2.2.1 Hash Function

Hash functions are used with other algorithms to provide a number of security services. A hash function is used in conjunction with a digital signature algorithm to compute a digital signature (see [FIPS 186-2]). A hash function may be used as part of a random number generator. A hash function is used with a key as part of its input to provide a Keyed-Hash Message Authentication Code (HMAC) (see Section 2.2.2.3.2 and [FIPS 198]). Approved hash functions are defined in [FIPS 180-2].

A hash function takes an input of arbitrary length and outputs a fixed size value. Common names for the output of a hash function are hash value, hash, message digest, and digital fingerprint. The maximum number of input bits is determined by the design of the hash function, as is the size of the output. All Approved hash functions are cryptographic hash functions. With a well-designed hash function, it is not feasible to find two messages that produce the same hash value.

Four hash functions are approved for Federal Government use and are defined in [FIPS 180-2]: SHA-1, SHA-256, SHA-384 and SHA-512. The hash size produced by SHA-1 is 160 bits, 256 bits for SHA-256, 384 bits for SHA-384, and 512 bits for SHA-512. [Note: it is anticipated that FIPS 180-2 will be approved in the near future.]

2.2.2.2 Symmetric Key Algorithms used for Encryption and Decryption

Encryption is used to provide confidentiality for data. The data to be protected is called plaintext when in its original form. Encryption transforms the data into ciphertext. Ciphertext can be

transformed back into plaintext using decryption. The approved algorithms for encryption/decryption are symmetric key algorithms: AES and TDES. Each of these algorithms operates on blocks (chunks) of data during a single encryption or decryption operation. For this reason, these algorithms are commonly referred to as block cipher algorithms.

2.2.2.2.1 Advanced Encryption Standard (AES)

The AES algorithm is specified in [FIPS 197]. AES encrypts and decrypts data in 128-bit blocks, using either 128, 192 or 256 bit keys. The nomenclature for AES for the different key sizes is AES-*x*, where *x* is the key size. All three key sizes are considered adequate for Federal Government applications.

2.2.2.2.2 Triple DES (TDES)

Triple DES is defined in [ANSI X9.52] and has been adopted in [FIPS 46-3]. TDES encrypts and decrypts data in 64-bit blocks, using three 56-bit keys. It is **RECOMMENDED** that Federal applications use three distinct keys. [ANSI X9.52] specifies seven modes of operation that are used with TDES for encryption.

2.2.2.2.3 Modes of Operation

With a block cipher algorithm, the same plaintext block will always encrypt to the same ciphertext block whenever the same key is used. If the multiple blocks in a typical message were to be encrypted separately, an adversary could easily substitute individual blocks, possibly without detection. Furthermore, data patterns in the plaintext would be apparent in the ciphertext.

Cryptographic modes of operation have been defined to alleviate this problem by combining the basic cryptographic algorithm with some sort of feedback of the information derived from the cryptographic operation. The NIST Recommendation for Block Cipher Modes of Operation [SP 800-38A] defines modes of operation for the encryption and decryption of data using block cipher algorithms such as AES and TDES. [ANSI X9.52], which was adopted by [FIPS 46-3], defines three additional TDES modes that pipeline or interleave the cryptographic operations to improve the efficiency of TDES.

2.2.2.3 Message Authentication Codes (MACs)

Message Authentication Codes (MACs) provide data authentication and integrity. A MAC is a cryptographic checksum on the data that is used to provide assurance that the data has not changed and that the MAC was computed by the expected entity. Although message integrity is often provided using non-cryptographic techniques known as message detection codes, these codes can be altered by an adversary to effect an action to the adversary's benefit. The use of an Approved cryptographic mechanism, such as a MAC, can alleviate this problem. In addition, the MAC can provide a recipient with assurance as to the identity of the originator of any data for which a MAC is provided.

The computation of a MAC requires the use of a secret key that **SHOULD** be known only by the party that generates the MAC and by the intended recipient(s) of the MAC, and the data on which the MAC is calculated. Two types of algorithms for computing a MAC have been approved: MAC algorithms that are based on block cipher algorithms, and MAC algorithms that are based on hash functions.

2.2.2.3.1 MACs Using Block Cipher Algorithms

[SP 800-38B] defines modes to compute a MAC using Approved block cipher algorithms such as AES and TDES. The key and block size used to compute the MAC depends on the algorithm used. [Note: SP 800-38B is under development. This section will be revised after SP 800-38B is published.]

2.2.2.3.2 MACs Using Hash Functions

[FIPS 198], *Keyed-Hash Message Authentication Code (HMAC)*, specifies the computation of a MAC using an Approved hash function. The algorithm requires a single pass through the entire data. A variety of key sizes are allowed for HMAC; the choice of key size depends on the amount of security to be provided to the data and the hash function used. See Section 3.3 for guidance in the selection of key sizes.

2.2.2.4 Digital Signature Algorithms

Digital signatures are used to provide authentication, integrity and non-repudiation. Digital signatures are used in conjunction with hash algorithms and are computed on data of any length (up to a limit that is determined by the hash algorithm). [FIPS 186-2] specifies algorithms that are approved for the computation of digital signatures; this standard defines the Digital Signature Algorithm (DSA) and adopts two algorithms specified in ANSI standards: [ANSI X9.31] (RSA and Rabin-Williams) and [ANSI X9.62] (the Elliptic Curve Digital Signature Algorithm - ECDSA). [A revision for FIPS 186-2 is planned. At that time, this section may be revised.]

2.2.2.4.1 DSA

The Digital Signature Algorithm (DSA) is specified in [FIPS 186-2] and will be revised as FIPS 186-3 in the near future. The revised standard will define this algorithm for specific key sizes[1]: 1024, 2048, 3072, 7680 and 15,360 bits. The revised DSA will produce digital signatures[2] of 320, 448, 512, 768 or 1024 bits. Older systems (legacy systems) used smaller key sizes. While it may be appropriate to continue to verify and honor signatures created using these smaller key sizes, new signatures **SHOULD NOT** be created using these key sizes.

2.2.2.4.2 RSA

[This may need to be revised, depending on how FIPS 186-3 deals with X9.31 and PKCS#1.]

The RSA algorithm, as specified in [ANSI X9.31] and [PKCS #1] (version 1.5 and higher), has been adopted for the computation of digital signatures in [FIPS 186-3]. [ANSI X9.31] defines key sizes[3] of 1024 bits or more in increments of 256 bits; [PKCS #1] does not specify key sizes. An additional difference between [ANSI X9.31] and [PKCS #1] is that each specifies a different format for constructing the data to be signed. RSA produces digital signatures that are one bit in length less than the key size. Older systems (legacy systems) used smaller key sizes. While it

---

[1] For DSA, the key size is considered to be the size of the modulus *p*. Another value, *q*, is also important when defining the security afforded by DSA.

[2] The length of the digital signature is twice the size of *q* (see the previous footnote).

[3] For RSA, the key size is considered to be the size of the modulus *n*.

may be appropriate to continue to verify and honor signatures created using these smaller key sizes, new signatures **SHOULD NOT** be created using these key sizes.

2.2.2.4.3 ECDSA

The Elliptic Curve Digital Signature Algorithm (ECDSA), as specified in [ANSI X9.62], has been adopted for the computation of digital signatures in [FIPS 186-2]. [ANSI X9.62] specifies a minimum key size[4] of 160 bits. ECDSA produces digital signatures that are twice the length of the key size. Recommended elliptic curves are provided in [FIPS 186-2].

2.2.2.5 Key Establishment Algorithms

Key establishment algorithms are used to set up keys to be used between communicating entities. Two types of key establishment are defined: key transport and key agreement. Approved key establishment schemes are provided in [FIPS XXX].

Key transport is the distribution of a key (and other keying material) from one entity to another entity. The keying material is usually encrypted by the sending entity and decrypted by the receiving entity(ies). If a symmetric algorithm (e.g., AES) is used to encrypt the keying material to be distributed, the sending and receiving entities need to know the secret key encrypting key. In this case, the key transport algorithm is commonly known as a key wrapping algorithm. If a public key algorithm is used to distribute the keying material, a key pair is used as the key encrypting key. In this case, the sending entity encrypts the keying material using the receiving entity's public key; the receiving entity decrypts the received keying material using the associated private key.

Key agreement is the participation by both entities (i.e., the sending and receiving entities) in the creation of shared keying material. This is accomplished using public key techniques. Each entity has either a static key pair or an ephemeral key pair or both.

[FIPS XXX] adopts selected key establishment schemes defined in ANSI standards that use public key algorithms: [ANSI X9.42], [ANSI X9.44], and [ANSI X9.63]. [ANSI X9.42] specifies key agreement schemes, and [ANSI X9.44] and [ANSI X9.63] specify both key agreement and key transport schemes. [FIPS XXX] also specifies techniques for key wrapping.

2.2.2.5.1 Discrete Log Key Agreement Schemes (Finite Field Arithmetic)

Key agreement schemes based on the intractability of the discrete logarithm problem and using finite field arithmetic have been adopted in [FIPS XXX] from ANSI [X9.42]. [FIPS XXX] has adopted seven of the eight schemes defined in [ANSI X9.42]. Each scheme provides a different configuration of required key pairs that may be used, depending on the requirements of a communication situation.

2.2.2.5.2 RSA Key Transport

RSA key transport schemes have been adopted from [ANSI X9.44]. [Further text to be developed, depending on the contents of FIPS XXX.]

---

[4] For elliptic curves, the key size is the length *f* of the order *n* of the base point *G* of the chosen elliptic curve.

2.2.2.5.3 Elliptic Curve Key Agreement and Key Transport

Elliptic curve key agreement and key transport schemes based on the intractability of the discrete logarithm problem and using elliptic curve arithmetic have been adopted in [FIPS XXX] from [ANSI X9.63]. Seven of the eleven key agreement schemes have been adopted. [Further text will be added as FIPS XXX is developed further.]

2.2.2.5.4 Key Wrapping

Key wrapping is the encryption of a key by a key encryption key using a symmetric algorithm (e.g., an AES key is encrypted by an AES key encrypting key). [Further text will be added as FIPS XXX is developed further.]

2.2.2.6 Random Number Generation

Random number generators (RNGs) are required for the generation of keying material (e.g., keys and IVs). Two classes of RNGs are defined: deterministic and non-deterministic. Deterministic RNGs, often called pseudorandom number generators, use cryptographic algorithms and the associated keying material to generate random numbers; non-deterministic RNGs, often called true RNGs, produce output that is dependent on some unpredictable physical source that is outside human control.

Pseudorandom number generation (PRNG) algorithms defined in [FIPS 186-2] (including those defined in [ANSI X9.31] and [ANSI X9.62][5]) use either hash functions or AES to generate random numbers that are used for cryptographic applications (e.g., key or IV generation). PRNGs are initialized with a starting value, called a seed. The seed may be public or secret, depending on its use.

2.2.3 Cryptographic Information

Several different classes of keys are used by the Approved algorithms. Many of these keys are associated with other keying material. The keying material **SHOULD** be protected. In addition, control information and the results from applying the protection mechanisms (e.g., the ciphertext or MAC) **SHOULD** be protected. Suggested protection mechanisms are also provided.

2.2.3.1 Protection Requirements

This section defines the security and association protections that **SHOULD** be provided to cryptographic information (i.e., cryptographic keys, other keying material and other cryptographic-related information) during its lifetime. Section 2.2.3.2 provides guidance on how the protection can be provided.

2.2.3.1.1 Cryptographic Keys

Several different classes of keys, grouped according to function and according to their useful life span (life cycle), are defined. The life cycle of each type of key is further discussed in Section 2.3.

1. *Signing private key*: Signing private keys are the private keys of a key pair that are used by public key algorithms to generate digital signatures with possible long-term

[5] As allowed in FIPS 186-2.

implications. When properly handled, signing private keys can be used to provide authentication, integrity and non-repudiation.

2. *Signature verification public key*: A signature verification public key is the public key of a key pair that is used by a public key algorithm to verify digital signatures, either for non-repudiation purposes, to determine the integrity of data, to authenticate a user's identity, or a combination thereof.
3. *Secret authentication key*: Secret authentication keys are used with symmetric key algorithms to provide assurance of the integrity and source of messages or communication sessions.
4. *Authentication private key*: An authentication private key is used with a public key algorithm to provide assurance as to the integrity of information, and the identity of the originating entity or the source of messages or communication sessions. Non-repudiation is not necessary for a private key used only for authentication.
5. *Authentication public key*: An authentication public key is used with a public key algorithm to determine the integrity of information and to authenticate the identity of entities, or the source of messages or communication sessions.
6. *Long-term data encryption key*: These keys are symmetric (secret) keys that are used with symmetric algorithms over a long period of time to apply confidentiality protection on information. Keys used for the encryption of other keys are discussed below.
7. *Short-term data encryption key:* These keys are symmetric (secret) keys that are used with symmetric algorithms over a short period to apply confidentiality protection on information; an example of such a short period would be a communication session or a single message. Keys used for the encryption of other keys are discussed below. These data encryption keys are generated as needed.
8. *Random number generation key*: These keys are symmetric (secret) keys used with a symmetric algorithm to generate pseudorandom numbers.
9. *Long-term key encrypting key used for key wrapping*: Key encrypting keys used for key wrapping are used with symmetric key algorithms. This long-term key encrypting key encrypts either data encrypting keys or other key encrypting keys over a relatively long period of time.
10. *Short-term key encrypting key used for key wrapping*: Key encrypting keys used for key wrapping are used with symmetric key algorithms. This key encrypting key encrypts either short-term data encrypting keys or other short-term key encrypting keys over a relatively short period of time.
11. *Master key used for key derivation*: A "master key" is used to derive other keying material. A master key could be either a secret (symmetric) key or a key pair; however, this guideline only addresses the case where the master key is a secret (symmetric) key.
12. *Key derived from a master key*: These keys **SHOULD** be protected in accordance with their use. A derived key could be either a secret (symmetric) key or a key pair; however, this guideline only addresses the case where the derived key is a secret (symmetric) key.

13. *Key transport private key*: Key transport private keys are used to decrypt keys that have been encrypted by the associated public key using a public key algorithm. They are usually used to establish multiple keys (e.g., key encrypting keys used for key wrapping, data encrypting keys or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).
14. *Key transport public key*: Key transport public keys are used to encrypt keys using a public key algorithm. They are used to establish keys (e.g., key encrypting keys used for key wrapping, data encrypting keys or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).
15. *Secret key agreement key*: These keys are used to establish keys (e.g., key encrypting keys used for key wrapping, data keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).
16. *Static key agreement private key*: Static private keys used for key agreement are used to establish keys (e.g., key encrypting keys used for key wrapping, data keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).
17. *Static key agreement public key*: Static public keys used for key agreement are used to establish keys (e.g., key encrypting keys used for key wrapping, data keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).
18. *Ephemeral key agreement private key*: Ephemeral private keys used for key agreement are used only once to establish one or more keys (e.g., key encrypting keys used for key wrapping, data keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).
19. *Ephemeral key agreement public key*: Ephemeral public keys used for key agreement are used only once to establish one or more keys (e.g., key encrypting keys used for key wrapping, data keys, or MAC keys) and, optionally, other keying material (e.g., Initialization Vectors).
20. *Secret authorization key*: Secret authorization keys are used to provide privileges to an entity. The key is known by the access authority and an entity seeking access to resources.
21. *Authorization private key*: An authorization private key is used to provide privileges to an entity. The key is known only by an entity seeking access to resources.
22. *Authorization public key*: An authorization public key is used to verify privileges for an entity that knows the associated private key.

Table 1 provides a summary of the protection requirements for these keys during distribution and storage. Methods for providing the necessary protection are discussed in Section 2.2.3.2.

Guide to Table 1:

- Column 1 (Key Type) identifies the key types.
- Column 2 (Security Service) indicates the type of security service that may be provided by the key.

- Column 3 (Security Protection) indicates the keys that require confidentiality or integrity protection or for which long-term availability may be required.
- Column 4 (Association Protection) indicates those keys for which the association with either the usage or application, the other entity or other indicated information **SHOULD** be protected.
- Column 5 (Validation) indicates the keys that need to be validated as defined in [FIPS XXX] and in this guideline. See Section ??? for further details.
- Column 6 (Period of Protection) indicates the length of time that the key needs to be protected.

Table 1: Protection requirements for cryptographic keys.

Key Type	Security Service	Security Protection	Association Protection	Validation	Period of Protection
Signing private key	Authentication; Integrity; Non-repudiation	Confidentiality; Integrity	Usage or application; Domain parameters; Signature verification public key;		For the cryptoperiod of the private key, and until the private key is destroyed
Signature verification public key	Authentication; Integrity; Non-repudiation	Integrity; Long-term availability	Usage or application; Other entity; Domain parameters; Signing private key; Signed data	For association with signing private key	As long as signed data may need to be verified
Secret authentication key	Authentication; Integrity	Confidentiality; Integrity; Long-term availability	Usage or application; Other entity; Authenticated data		As long as the authenticated data may need to be authenticated, and until the authentication key is destroyed
Authentication private key	Authentication; Integrity	Confidentiality; Integrity	Usage or application; Authentication public key; Domain parameters		For the cryptoperiod of the private key, and until the private key is destroyed
Authentication public key	Authentication; Integrity	Integrity; Long-term availability	Usage or application; Other entity; Authenticated data; Authentication private key	For association with private key	As long as the authenticated data may need to be authenticated
Long-term	Confidentiality	Confidentiality;	Usage or		As long as the encrypted

Key Type	Security Service	Security Protection	Association Protection	Validation	Period of Protection
data encryption key		Integrity; Long-term availability	application; Other entity; Encrypted data		data may need to be decrypted, and until the data encryption key is destroyed
Short-term data encryption key	Confidentiality	Confidentiality; Integrity[6]; Long term availability	Encrypted data		As long as the encrypted data may need to be decrypted, and until the data encryption key is destroyed
RNG key	Support	Confidentiality; Integrity	Usage or application		Until no longer needed to generate random numbers and until destroyed
Long-term key encrypting key used for key wrapping	Support	Confidentiality; Integrity; Long-term availability	Usage or application; Other entity; Encrypted keys		As long as encrypted keys may need to be decrypted, and until the key encrypting key is destroyed
Short-term key encrypting key used for key wrapping	Support	Confidentiality; Integrity[7]	Usage or application; Other entity; Encrypted keys		As long as encrypted keys may need to be decrypted, and until the key encrypting key is destroyed
Master key used for key derivation	Support	Confidentiality; Integrity; Long-term availability	Usage or application; Other entity; Derived keys		For the lifetime of any keys derived using this key, and the master key is destroyed
Key derived from a master key	Depends on key use	Integrity; Otherwise depends on use	Master key and protected data; Otherwise depends on use		Depends on the use of the derived key; until the key is destroyed
Key transport private key	Support	Confidentiality; Integrity	Usage or application; Encrypted keys; Key transport public		As long as the transported key needs to be decrypted

[6] Although short term and ephemeral keys are normally used during a very short period (e.g., during one communication session), they require integrity protection from the time that they are generated until the keys are no longer retained. This period of retention may be long enough for an adversary to substitute a key known to the adversary without detection unless integrity protection is provided.

[7] Although short term and ephemeral keys are normally used during a very short period (e.g., during one communication session), they require integrity protection from the time that they are generated until the keys are no longer retained. This period of retention may be long enough for an adversary to substitute a key known to the adversary without detection unless integrity protection is provided.

Key Type	Security Service	Security Protection	Association Protection	Validation	Period of Protection
			key		
Key transport public key	Support	Integrity; Long-term availability	Usage or application; Other entity; Key transport private key	Yes	For the cryptoperiod of the public key
Secret key agreement key	Support	Confidentiality; Integrity; Long-term availability	Usage or application; Other entity		Until no longer needed to determine a key, and the key agreement key is destroyed; normally, this is the cryptoperiod of the key agreement key, but may be longer if a key needs to be reconstructed during key recovery
Static key agreement private key	Support	Confidentiality; Integrity; Long-term availability	Usage or application; Domain parameters; Static key agreement public key		Until no longer needed to determine a key, and the private key is destroyed; normally, this is the cryptoperiod of the private key, but may be longer if a key needs to be reconstructed during key recovery
Static key agreement public key	Support	Integrity; Long-term availability	Usage or application; Other entity; Domain parameters; Static key agreement private key	Yes	Until no longer needed to determine a key
Ephemeral key agreement private key	Support	Confidentiality; Integrity			Until the key agreement process is complete, and the ephemeral key is destroyed
Ephemeral key agreement public key	Support	Integrity[8]		Yes	Until the key agreement process is complete
Secret authorization keys	Authorization	Confidentiality; Integrity	Usage or application; Other entity		For the cryptoperiod of the authorization key, and until the authorization key is

[8] Although short term and ephemeral keys are normally used during a very short period (e.g., during one communication session), they require integrity protection from the time that they are generated until the keys are no longer retained. This period of retention may be long enough for an adversary to substitute a key known to the adversary without detection unless integrity protection is provided.

Key Type	Security Service	Security Protection	Association Protection	Validation	Period of Protection
					destroyed
Authorization private key	Authorization	Confidentiality; Integrity	Usage or application; Authorization public key		For the cryptoperiod of the private key, and until the private key is destroyed
Authorization public key	Authorization	Integrity	Usage or application; Other entity; Authorization private key	Yes	For the cryptoperiod of the key

2.2.3.1.2 Other Keying Material

Other information used in conjunction with cryptographic algorithms also **SHOULD** be protected.

1. *Domain Parameters*: Domain parameters are used in conjunction with some public key algorithms to generate key pairs or to create digital signatures (i.e., DSA, and the Diffie-Hellman and MQV key agreement schemes specified in [FIPS XXX]).
2. *Initialization Vectors*: Initialization vectors (IVs) are used by several modes of operation for encryption and decryption (see Section 2.2.2.2.3) and for the computation of MACs using block cipher algorithms (see Section 2.2.2.3.1).
3. *Shared Secrets:* Shared secrets are generated during a key establishment process as defined in [FIPS XXX].
4. *Secret seeds*: Secret seeds are used in the generation of *pseudorandom* numbers that **SHOULD** remain secret (e.g, used to generate keying material that must remain secret or private).
5. *Public seeds*: Public seeds are used in the generation of pseudorandom numbers that may be validated (e.g., elliptic curve domain parameters). A public seed is sometimes called “salt”.
6. *Nonce*: A nonce is a non-repeating value that is used only once for a particular purpose. A nonce is often used to prevent a replay and other types of attacks. A nonce may be used in certain modes of operation to generate an IV.
7. *Intermediate Results*: The intermediate results of cryptographic operations **SHOULD** be protected.

Table 2 provides a summary of the protection requirements for this material during distribution and storage. Methods for providing the necessary protection are discussed in Section 2.2.3.2.

Guide to Table 2:

- Column 1 (Keying Material Type) identifies the type of keying material.

- Column 2 (Security Service) indicates the type of security service for which the keying material may be used.
- Column 3 (Security Protection) indicates the keying material that requires confidentiality or integrity protection or for which long-term availability may be required.
- Column 4 (Association Protection) indicates the keying material for which the association with the usage or application, the other entity or other indicated information **SHOULD** be protected.
- Column 5 (Validation) indicates the keying material that needs to be validated as defined in [FIPS XXX] and in this guideline. See Section 3.3 for further details.
- Column 6 (Period of Protection) indicates the length of time that the keying material needs to be protected.

Table 2. Protection requirements for other keying material.

Keying Material Type	Security Service	Security Protection	Association Protection	Validation	Period of Protection
Domain parameters	Depends on key assoc. with the params.	Integrity; Long-term availability	Usage or application; Private and public keys	Yes	Until no longer needed to generate keys, or verify signatures
Initialization vectors	Depends on algorithm	Integrity; Long-term availability	Protected data		Until no longer needed to process the protected data
Shared secrets	Support	Integrity; Long-term availability?	Generated data?		Until no longer needed to generate keying material, and the shared secret is destroyed
Secret seeds	Support	Confidentiality; Integrity?	Usage or application?		Until no longer needed to generate keying material and the shared secret is destroyed
Public seeds	Support	Integrity	User or application; Generated data		Until no longer needed to process generated data
Nonce	Support	Integrity; Long-term availability?	User or application; Other entity; Data processed using the nonce	That is has not been repeated[9]	Until no longer needed to process data using the nonce
Intermediate results	Support	Confidentiality; Integrity	Usage or application		Until no longer needed and the intermediate results are destroyed

[9] Only when generated in an orderly fashion (e.g., when a counter is used); validation may not be practical if the nonce is generated randomly.

2.2.3.1.3 Other Information

Certain information that is created as the result of applying cryptographic mechanisms **SHOULD** also be protected.

1. *Hash value/hash/message digest*: The hash value that is computed on data using a mechanism identified in Section 2.2.2.1 **SHOULD** be protected, if retained. Note that for many applications, such as the computation of digital signatures, the hash value may not be retained; the hash value is considered to be the intermediate result of creating or verifying a digital signature.
2. *Encrypted data*: The data as encrypted by a cryptographic mechanism in Section 2.2.2.2 **SHOULD** be protected to ensure that it will decrypt correctly.
3. *MAC*: The message authentication code that is computed using a mechanism from Section 2.2.2.3 **SHOULD** be protected to assure that it can be used to determine the authenticity and integrity of the data upon which the MAC is computed.
4. *Digital signature*: The digital signature that is computed using a mechanism from Section 2.2.2.4 **SHOULD** be protected to assure that it can be used to determine the authenticity, integrity and susceptibility to non-repudiation of the data upon which the digital signature is computed.
5. *Key control information*: Information related to the keying material (e.g., the identity, purpose, or a counter) **SHOULD** be protected to ensure that the associated keying material can be correctly used.
6. *Random numbers*: The random numbers created by a random number generator **SHOULD** be protected when retained. When used directly as keying material, the random numbers **SHOULD** be protected as discussed in Sections 2.2.2.3.1 and 2.2.2.3.2. Otherwise, the random numbers **SHOULD** be protected as specified in Table 3.
7. *Passwords*: A password is used to acquire access to privileges. As such, it is used as an authentication mechanism.
8. *Audit information*: Audit information that contains key management events **SHOULD** be protected.

Table 3 includes a summary of the protection mechanisms required for this information. Methods for providing the protection mechanisms are discussed in Section 2.2.3.2.

Guide to Table 3:

- Column 1 (Information) identifies the type of information to be protected.
- Column 2 (Security Service) indicates the type of security service with which the information is used.
- Column 3 (Security Protection) indicates the types of information that require confidentiality or integrity protection or for which long-term availability may be required.
- Column 4 (Association Protection) indicates the other information or keying material with which the information listed in column 1 **SHOULD** be associated.

- Column 5 (Period of Protection) indicates the length of time that the information needs to be protected.

Table 3. Protection requirements for other information.

Information	Security Service	Security Protection	Association Protection	Period of Protection
Hash value	Integrity; Support	Integrity; Confidentiality? Long-term availability?	Hashed data	Until the hash value no longer needed
Encrypted data	Confidentiality	Integrity; Long-term availability	Data encryption key	Until no longer needed to be decrypted
MAC	Integrity; Authentication; Authorization	Integrity; Long-term availability	Authenticated data; Authentication key;	Until no longer needed to authenticate the authenticated data
Digital signature	Integrity; Authentication; Non-repudiation	Integrity; Long-term availability	Signed data; Signing private key	Until the signed data no longer needs to be verified
Key control information (e.g., IDs, purpose)	Support	Integrity; Long-term availability	Key	Until the associated key is destroyed
Random number	Support	Confidentiality? Integrity		Until no longer needed, and the random number is destroyed
Password	Authentication	Confidentiality; Integrity; Long-term availability?	Usage or application; Owning entity	Until replaced or no longer needed to authenticate the entity
Audit information	Support	Access authorization; Integrity; Long-term availability	Audited events; Key control information	Until no longer needed

2.2.3.2 Protection Mechanisms

During the lifetime of cryptographic information, the information is either “in transit” (e.g., is in the process of being manually or electronically distributed to another entity for use by that entity) or is “at rest” (e.g., the information is in storage). In either case, the keying material **SHOULD** be protected in accordance with Section 2.2.3.1. However, the choice of protection mechanisms may vary. Although several methods of protection are provided in the following subsections, not all methods provide equal security. The method selected **SHOULD** be carefully selected.

2.2.3.2.1 Protection Mechanisms for Cryptographic Information in Transit

Cryptographic information in transit may be keying material being distributed in order to establish a key (see Section 2.3.1.5), or cryptographic information being backed up or archived for possible use or recovery in the future (see Sections 2.3.2.2 and 2.3.3.1). This may be accomplished using "manual methods" (i.e., the information is in hard copy or on an electronic media, such as a CD-ROM), or using electronic communication protocols. For some protocols, the protections are provided by the protocol; in other cases, the protection for the keying material MUST be provided directly on the keying material. It is the responsibility of the originating entity to apply protection mechanisms, and the responsibility of the recipient to undo or check the mechanisms used.

2.2.3.2.1.1 Confidentiality

Keying material sometimes requires confidentiality protection during transit. The keying material **SHOULD** be protected using one or more of the following mechanisms:

- Manual method:

  (1) The keying material is encrypted,

  -OR-

  (2) The keying material is separated into key components. Each key component is handled so that no single individual can acquire access to all key components.

  -OR-

  (3) Appropriate physical and procedural protection is provided (e.g., by using a trusted courier).

- Electronic distribution via communication protocols: The keying material is encrypted.

2.2.3.2.1.2 Integrity

Integrity protection consists of both the detection of modifications to the information and the restoration of the information when a modification is detected. The integrity of cryptographic information during transit **SHOULD** be protected using one or more of the following mechanisms:

- Manual method (physical protection is provided):

  (1) An integrity mechanism (e.g., MAC, digital signature or CRC) is used on the information, and the resulting code (e.g., MAC or digital signature) is provided to the recipient. Note: the MAC or digital signature itself may not require further integrity protection.

  -OR-

  (2) The information is used to perform the appropriate cryptographic operation. If the use of the keying material produces incorrect results, the received information is incorrect.

- Electronic distribution via communication protocols (provided by the user or by the communication protocol):

(1) A cryptographic integrity mechanism (e.g., a MAC or digital signature) is used on the information, and the resulting code (e.g., MAC or digital signature) is provided to the recipient. The integrity mechanism may be applied only to the cryptographic information, or may be applied to an entire message,

-OR-

(2) The information is used to perform the appropriate cryptographic operation. If the use of the keying material produces incorrect results, the received information is incorrect.

If an error is detected in the received information, and the receiver requires that the information be entirely correct (e.g., the receiver cannot proceed when the information is in error), then:

- the information **SHOULD NOT** be used,
- the information **SHOULD** be destroyed in accordance with Section 2.3.3.3, and
- the recipient **SHOULD** request that the information be resent.

2.2.3.2.1.3 Long-term Availability

Long-term availability is not an issue for cryptographic information in transit.

2.2.3.2.1.4 Association with Usage or Application

The association of keying material with its usage or application **SHOULD** be either specifically identified during the distribution process, or be implicitly defined by the use of the application.

2.2.3.2.1.5 Association with the other Entity

The association of keying material with the appropriate entity **SHOULD** be either specifically identified during the distribution process, or be implicitly defined by the use of the application.

2.2.3.2.1.6 Association with Other Information

Any association with other related information (e.g., domain parameters, the encryption/decryption key or IVs) **SHOULD** be either specifically identified during the distribution process, or be implicitly defined by the use of the application.

2.2.3.2.2 Protection Mechanisms for Information in Storage

Cryptographic information that is not in transit is at rest in some device or storage media. This may include copies of the information that is also in transit. This information **SHOULD** be protected in accordance with Section 2.2.3.1. A variety of protection mechanisms may be used.

The cryptographic information may be stored so as to be immediately available to an application (e.g., on a local hard disk or a server); this would be typical for keying material stored within the cryptographic module or in immediately accessible storage (e.g., on a local hard drive). The keying material may also be stored in electronic form on a removable media (e.g., a CD-ROM), in a remotely accessible location, or in hard copy form and placed in a safe; this would be typical for backup or archive storage.

2.2.3.2.2.1 Confidentiality

One or more of the following mechanisms **SHOULD** be used to provide confidentiality for keying material in storage:

(1) The keying material resides in an Approved cryptographic module (cryptomodule). The cryptomodule is designed to be compliant with FIPS 140-2 level 2 (or higher) and tested by an accredited laboratory of the Cryptographic Module Validation Program (CMVP). Information on this program is available at http://csrc.nist.gov/cryptval.

(2) The keying material resides in an Approved cryptographic module (cryptomodule). The cryptomodule is designed to be compliant with FIPS 140-2 and has been tested by an accredited laboratory of the Cryptographic Module Validation Program (CMVP). When the cryptomodule is not is use, it is physically protected (e.g., by a locked door).

(3) The keying material is encrypted and stored off-line in a secure environment (e.g., in a safe with limited access).

(4) The keying material is stored off-line in a secure environment (e.g., in a safe with limited access) with access available using dual control procedures (i.e., two or more individuals are required for access to the keying material). It is the responsibility of each individual to assure that the confidentiality of the keying material is maintained.

(5) Keying material is split into multiple key components. The key components are combined in a secure environment (e.g., in an Approved cryptomodule) when constructing the keying material. The key components are stored separately, under split knowledge procedures whereby no single individual has access to more than one key component. Each key component is protected as sensitive information (e.g., the key component is not made publicly available). One method for splitting keying material into two components, for example, is to generate a random bit string of the intended length of the components, using this bit string as one of the components, and exclusive Or-ing the bit string to the generated keying material to create the second component. However, other methods for splitting and combining components are allowed.

2.2.3.2.2.2 Integrity

Integrity protection is concerned with ensuring that the information is correct. Absolute protection against modification is not possible. The best that can be done is to use reasonable measures to prevent modifications, to use methods to detect (with a very high probability) any modifications that occur, and to restore the information to its original content when modifications have been detected.

All cryptographic information requires integrity protection. It is **RECOMMENDED** that the prevention of modifications be assisted by storing the information:

(1) In an Approved cryptographic module or operating system that limits access to the stored information,

(2) In a computer system or on a media that is not connected to other systems,

(3) Outside a computer system in a physically secure environment (e.g., in a safe with limited access) with appropriate access controls.

In addition, it is **RECOMMENDED** that one or more of the following mechanisms be used to detect modifications to the information:

(1) A cryptographic integrity mechanism (e.g., MAC or digital signature) is used on the information, and the resulting code (e.g., MAC or digital signature) is available for verifying the integrity of the stored information.

(2) The information is used to perform the appropriate cryptographic operation. If the use of the keying material produces incorrect results, the stored information is incorrect.

If the cryptographic information needs to be restored when an error is detected, multiple copies of the information **SHOULD** be maintained in physically separate locations (i.e., in backup or archive storage; see Sections 2.3.2.2.1 and 2.3.3.1). The integrity of each copy **SHOULD** be periodically checked.

2.2.3.2.2.3 Long-term Availability

Cryptographic information may need to be readily available for as long as data is protected by the information. The primary method for providing this protection is to make one or more copies of the cryptographic information and store them in separate locations. During a key's cryptoperiod, keying material requiring long-term availability **SHOULD** be stored in both normal operational storage (see Section 2.3.2.1) and in backup storage (see Section 2.3.2.2.1). Cryptographic information that is retained after the end of a key's cryptoperiod **SHOULD** be placed in archive storage (see Section 2.3.3.1). This guideline does not preclude the use of the same storage media for both backup and archive storage.

Specifics on the long-term availability requirement for each key type are addressed for backup storage in Section 2.3.2.1, and for archive storage in Section 2.3.3.1.

Recovery of this cryptographic information for use in replacing cryptographic information that is lost (e.g., from normal storage), or in performing cryptographic operations after the end of a key's cryptoperiod is discussed in Sections 2.3.2.2.2 and 2.3.3.1, and in Appendix B.

2.2.3.2.2.4 Association with Usage or Application

Cryptographic information is used with a given cryptographic mechanism (e.g., digital signatures or key establishment) or with a particular application. Protection **SHOULD** be provided to ensure that the information is not used incorrectly (e.g., not only **SHOULD** the usage or application be associated with the keying material, but the integrity of this association **SHOULD** be maintained). This protection can be provided by separating the cryptographic information from that of other mechanisms or applications, or by appropriate labeling of the information. Section 2.2.3.2.3 addresses the labeling of cryptographic information.

2.2.3.2.2.5 Association with the Other Entity

Some cryptographic information needs to be correctly associated with another entity, and the integrity of this association **SHOULD** be maintained. For example, a symmetric (secret) key used for the encryption of information, or keys used for the computation of a MAC **SHOULD** be associated with the other entity(ies) that shares the key. Public keys **SHOULD** be correctly associated (bound) with the owner of the key pair.

The cryptographic information **SHOULD** retain its association during storage by separating the information by "entity" or application, or by properly labeling of the information. Section 2.2.3.2.3 addresses the labeling of cryptographic information.

2.2.3.2.2.6 Association with Other Information

An association may need to be maintained between protected information and the keying material that protected that information. In addition, keys may require association with other keying material (see Section 2.2.3.1).

The association is accomplished by storing the information together or providing some linkage or pointer between the information. Typically, the linkage between a key and the information it protects is accomplished by providing an identity for a key, storing the identity with the key in an identification/label, and storing the key's identity with the protected information. The association **SHOULD** be maintained for as long as the protected information needs to be processed.

Section 2.2.3.2.3 addresses the labeling of cryptographic information.

2.2.3.2.3 Labeling of Cryptographic Information

Labels may be used with cryptographic information to define the use of that information or to provide a linkage between cryptographic information.

2.2.3.2.3.1 Labels for Keys

A label may be used to provide information for the use of a key. Different applications may require different labels for the same key type, and different labels may be required for different key types. It is the responsibility of an implementer to select a suitable label for a key. When labels are used, it is **RECOMMENDED** that the label accompany a key (i.e., is typically stored or transmitted with a key) and contain some subset of the following information:

- Identity of the key
- Association with other keying material (e.g., a public and private key **SHOULD** be associated with each other)
- Identity of the key's owner or the sharing entity
- Cryptoperiod (e.g., start date and end date)
- Key type (e.g., signing private key, encryption key, master key)
- Application (e.g., purchasing, email)
- Counter
- Domain parameters (e.g., the domain parameters used by DSA or ECDSA, or a pointer to them)
- Modulus
- Key encrypting key (e.g., identity of key wrapping key, algorithm of key wrapping key, etc.)
- Integrity protection mechanism (e.g., key and algorithm used to provide cryptographic protection, and protection code (e.g., MAC, digital signature))

- Other information (e.g., length of the key, protection requirements, who has access rights, additional conditions for use)

2.2.3.2.3.2 Labels for Other Information

Cryptographic information other than keying material may need to "point to" the keying material that was used to provide the cryptographic protection for the information. The label may also contain other related cryptographic information. When labels are used, it is **RECOMMENDED** that the label accompany the information (i.e., is typically stored or transmitted with the information) and contain some subset of the following information:

- The type of information (e.g., encrypted data, signed data)
- Source of the information (e.g., the entity that sent the information)
- Application (e.g., purchasing, email)
- Identity of the key to be used to undo or check the cryptographic protection mechanism (e.g., the identity of the decryption key or the signature verification public key)
- IV or nonce
- Other associated cryptographic information (e.g., a MAC or hash value)
- Any other information (e.g., who has access rights)

2.3 Key Management Lifecycle

Cryptographic key management encompasses the entire life cycle of cryptographic keys and other keying material.

A single item of keying material (e.g., a key) has several phases during its life, though some of these phases may, in fact, be very short:

- Pre-operational: The keying material is not yet available for normal cryptographic operations.
- Operational: The keying material is available and in normal use.
- Post-operational: The keying material is no longer in normal use, but access to the material is possible.
- Obsolete/destroyed: The keying material is no longer available. All records of its existence may have been deleted.

The following subsections discuss the various functions of key management. Each function maps into one of the above four phases. A key management system may not have all identified functions, since some functions may not be appropriate. For some keys, one or more functions may be combined, or the functions may be performed in a different order. For example, it may be appropriate to combine backup and archive storage into one location.

When functions are combined, the keying material **SHOULD** be handled as though the functions are separate (e.g., if keying material is only to be retained during the key's cryptoperiod but no longer, the keying material **SHOULD** be removed at the end of the cryptoperiod and not retained for archive purposes).

[A diagram visualizing the relationships between the phases and functions may be inserted later.]

2.3.1 Pre-operational Phase

During the pre-operational phase of the key management lifecycle, keying material is not yet available for normal cryptographic operations.

2.3.1.1 User Registration

During user registration, an entity becomes an authorized member of a security domain. In this phase, a user or device name may be established to identify the member during future transactions. In particular, security infrastructures may associate the identification information with the entity's keys (see 2.3.1.6, Key Registration). The entity may also establish various attributes during the registration process, such as email addresses or role/authorization information. As with identity information, these attributes may be associated with the entity's keys by the infrastructure to support secure application-level security services.

As applications will depend upon the identity established during this process, it is crucial that the registration authority establish appropriate procedures for the validation of identity. Identity may be established through an in-person appearance at a registration authority, or may be established entirely out-of-band. The strength (or weakness) of a security infrastructure will often depend upon the identification process.

User and key registration may be performed separately, or in concert. If performed separately, the entity will generally establish a shared secret (e.g., a password, PIN, or HMAC key); the shared secret may be used to authenticate keys during the key registration step. If performed in concert, the user establishes their identity and performs keys registration in the same process. In this case, a shared secret is unnecessary.

2.3.1.2 System Initialization

System initialization involves setting up or configuring a system for secure operation. This may include algorithm preferences, the identification of trusted parties, and the definition of domain parameter policies and any trusted parameters (e.g., recognizing certificate policies or the identification of IP address validation servers).

2.3.1.3 User Initialization

User initialization consists of an entity initializing its cryptographic application (e.g., installing and initializing software or hardware). This involves the use or installation (see Section 2.3.1.5) of the initial keying material obtained during user registration. Examples include the installation of a key at a CA, trust parameters, policies, trusted parties, and algorithm preferences.

2.3.1.4 Keying Material Installation

The security of keying material installation is crucial to the security of a system. For this function, keying material is installed for operational use within an entity's software, hardware, system, application, cryptomodule, or device using a variety of techniques. Keying material is installed when the software, hardware, system, application, cryptomodule, or device is initially set up, when new keying material is added to the existing keying material, and when existing keying material is replaced (via rekeying, key update, or key derivation - see Section 2.3.2.3).

The process for the initial installation of keying material (e.g., by manual entry, electronic key loader, by a vendor during manufacture) **SHOULD** include the protection of the keying material

during entry into a software/hardware/system/application/cryptomodule/ device, take into account the requirements of FIPS 140-2 and its differing requirements based on levels of protection, and include any additional procedures that may be required.

Note: Should this section contain guidance on:

- issues related to the installation of additional keying material,
- issues related to replacing existing keying material that are not covered in Section 2.3.2.3,
- issues related to keying material installation during key recovery that are not addressed in Sections 2.3.2.2.2 or 2.3.3.1, or in Appendix B.

Many applications or systems are provided by the manufacturer with keying material that is be used to test that the newly installed application/system is functioning properly. This test keying material **SHOULD** be replaced prior to operational use by installing the initial operational keying material.

2.3.1.5 Key Establishment

Key establishment includes the generation of keying material by an entity for its own use (e.g., for the protection of stored information); and the generation and distribution, or the agreement of keying material for communication between entities. During this process, some of the keying material may be in transit (e.g., the keying material is being manually or electronically distributed). Other keying material may be retained locally. In either case, the keying material **SHOULD** be protected in accordance with Section 2.2.3.1.

An entity may be an individual (person), organization, device or process. When keying material is generated by an entity for its own use, and the keying material is not distributed among "sub-entities" (e.g., is not distributed among various individuals, devices or processes within an organization), one or more of the appropriate protection mechanisms for stored information in Section 2.2.3.2.2 **SHOULD** be used.

Keying material that is distributed between entities or between sub-entities of a single entity **SHOULD** be protected using one or more of the appropriate protection mechanisms specified in Section 2.2.3.2.1. Any keying material that is not distributed (e.g., the private key of a key pair, or one's own copy of a symmetric key) **SHOULD** be protected using one or more of the appropriate protection mechanisms specified in Section 2.2.3.2.2.

2.3.1.5.1 Generation and Distribution of Key Pairs

Key pairs **SHOULD** be generated in accordance with the mathematical specifications of the appropriate Approved standard.

A static key pair **SHOULD** be generated within a FIPS 140-2 validated cryptographic module[10]. A static key pair **SHOULD** be either generated by the entity that "owns" the key pair (i.e., the entity that uses the private key in the cryptographic computations) or by a facility that distributes the key pair in accordance with Section 2.3.1.5.1.3. When generated by the entity that owns the key pair, the signing private key **SHOULD NOT** be distributed to other entities. In the case of a

[10] Alternatively, a key generation may be performed in a facility that is approved for the generation of classified keying material.

signature verification public key and its associated private key, it is **RECOMMENDED** that the owner generate the keying material rather than any other entity generating the keying material for that owner; this will facilitate non-repudiation.

Ephemeral keys are often used for key establishment (see [FIPS XXX]). They are short-lived and are statistically unique to each execution of a key establishment process (e.g., unique to each message or session). It is **RECOMMENDED** that an ephemeral key pair be generated within a FIPS 140-2 validated cryptographic module. An ephemeral key pair **SHOULD** be generated by the owner of that key pair. The ephemeral private key **SHOULD NOT** be distributed to other entities.

The generated key pairs **SHOULD** be protected in accordance with Section 2.2.3.1.1.

[Note: The generation and distribution of key shares are not currently addressed in this guideline, since there are no Approved methods for key share generation. Should we include discussions on non-Approved cryptographic methods? We would need to provide a caveat that the method is not currently approved. This philosophy also needs to be addressed for other suggested subjects for this guideline (e.g., *k* of *n* systems, password based key generation).]

2.3.1.5.1.1 Distribution of Static Public Keys

Static public keys are relatively long lived and are typically used for a number of executions of an algorithm. The distribution of the public key **SHOULD** provide assurance to the receiver of that key that:

a) the true owner of the key is known (i.e., the owner of the key pair); this requirement may be disregarded if anonymity is acceptable. However, the strength of the cryptographic system and trust in the validity of the protected data depends, in large part, on the assurance of the public key owner' identity,

b) the purpose/usage of the key is known (e.g., RSA digital signatures or elliptic curve key agreement),

c) any parameters associated with the public key are known (e.g., domain parameters), and

d) the public key has been properly generated (e.g., the owner of the public key actually has the associated private key and/or the mathematical properties of the public key are correct (public key validation)).

Keys falling into this category are:

- *The signature verification public key,*
- *The authentication public key,*
- *The key transport public key,*
- *The static key agreement public key,* and
- *The authorization public key*.

2.3.1.5.1.1.1 Distribution of a Trusted Anchor's Public Key

The public key of a trusted CA, or trust anchor, is the foundation for all PKI-based security services. The trust anchor is not a secret, but the *authenticity* of the trust anchor is the crucial

assumption for PKI. Trust anchors may be obtained through many different mechanisms, providing different levels of assurance. The types of mechanisms that are provided may depend on the role of the user in the infrastructure. A user that is only a “relying party” – that is, a user that does not have keys registered with the infrastructure – may use different mechanisms than a user that possesses keys registered by the infrastructure. Trust anchors are frequently distributed as "self-signed" X.509 certificates, that is, certificates that are signed by the subject public key of the certificate.

Trust anchors are often embedded within an application and distributed with the application. For example, the installation of a new web browser typically includes the installation or update for the user’s trust anchor list. Operating systems often are shipped with "code signing" trust anchor public keys. The user relies upon the authenticity of the software distribution mechanism to ensure that only valid trust anchors are installed during installation or update. However, in some cases other applications may install trust anchor keys in web browsers.

Trust anchors in web browsers are used for several purposes, including validation of S/MIME e-mail certificates and web server certificates for "secure websites" that use the SSL/TLS protocol to authenticate the web server and provide confidentiality. Relying party users who visit "secure" websites that have a certificate not issued by a trust anchor CA may be given an opportunity to accept that certificate, either for a single session, or permanently. They **SHOULD** also be cautious about accepting certificates from unknown certification authorities so that they do not, in effect, inadvertently add new permanent trust anchors.

Roaming users may have particular concerns about trust anchors used by web browsers when they use systems in kiosks, libraries, Internet cafes, or hotels and systems provided by conference organizers to access "secure websites." The user than has no control over the trust anchors installed in the host system, and simply trusts the host systems to make sound choices of trust anchors. Trust anchor distribution through software installation does not require that the relying party user be a direct participant in the infrastructure. The user trusts the software distribution mechanism to avoid installation of Trojan horses. Extending this trust to cover trust anchors for that application may be reasonable, and allows the user to obtain trust anchors without any additional procedures.

Where a user is a direct participant in the infrastructure, additional mechanisms are usually available. The user interacts with the infrastructure to register its keys (e.g., to obtain certificates) and these interactions may be extended to provide trust anchor information. This allows the user to establish trust anchor information with approximately the same assurance that the infrastructure has in the user’s keys. In the case of a PKI:

- The initial distribution of the public key of a trusted CA, or trust anchor, **SHOULD** be performed in conjunction with the presentation of a requesting entity's public key to a registration authority (RA) or CA during the certificate request process. In general, the CA's public key, associated parameters, key use, and proof-of-possession are conveyed as a self-signed X.509 public key certificate. The certificate is digitally signed by the private key that corresponds to the public key within the certificate. While parameters, and proof-of-possession may be clearly established from the self-signed certificate, the CA's identity information cannot be verified from the certificate itself.
- The trusted process used to convey a requesting entity's public key and assurances to the RA or CA **SHOULD** also protect the trust anchor information conveyed to the requesting

entity. In cases where the requesting entity appears in person, the trust anchor information may be provided at that time. If a shared secret has been established during user registration (see Section 2.3.1.1), the trust anchor information may be supplied with the requesting entity's certificate. In cases where the RA or CA delegates the verification of the requesting entity's identity to another trusted process, the trust anchor information **SHOULD** be provided along with the unique, unpredictable information (see Section 2.3.1.1.2, bullet 3).

2.3.1.5.1.1.2 Submission to a Registration Authority or Certification Authority

Public keys may be provided to a Certification Authority (CA) or to a registration authority (RA) for subsequent certification by a CA. During this process, the RA or CA **SHOULD** be provided with the assurances listed in Section 2.3.1.5.1.1 by the owner of the key or an authorized representative (e.g., the firewall administrator). The assurances include the owner's identity, the key use, any parameters to be used, and validation information.

In general, the owner of the key is identified in terms of an identity established during user registration (see 2.3.1.1). The key owner identifies the appropriate uses for the key, along with the parameters and any validation information. In cases where anonymous ownership of the public key is acceptable, the owner or the registration authority generates a pseudonym to be used as the identity.

In addition to the public key and the assurances listed above, proof of possession (POP) of the corresponding private key **SHOULD** be provided by the reputed owner of the key pair. If the owner is not required to perform POP, it is possible that the CA would bind the public key to the wrong entity. As a general rule, the owner **SHOULD** prove possession by performing operations with the private key that satisfy the indicated key use. For example, if a key pair is intended to support key transport, the owner **SHOULD** decrypt a key provided to the owner by the CA that is encrypted using the owner's public key. If the owner can correctly decrypt the ciphertext key using the associated private key, then the owner has established POP. In this case, POP **SHOULD NOT** be afforded by a generating and verifying digital signature with the key pair.

As with user registration, the strength of the security infrastructure depends upon the methods used for distributing the key to a RA or CA. There are many different methods, each appropriate for some range of applications. Some examples of common methods are:

- The public key, assurances and POP are provided by the public key owner in person, or by an authorized representative of the private key owner.
- The identity of the public key owner is established at the RA or CA in person or by an authorized representative of the public key owner during user registration. Unique, unpredictable information (e.g., an authenticator or cryptographic key) is provided at this time by the RA or CA to the owner as a shared secret. The public key, remaining assurances (key use, parameters, and validation information), and POP are provided to the RA or CA using or a communication protocol protected by the shared secret. The shared secret **SHOULD** be destroyed by the key owner as specified in Section 2.3.3.2 upon receiving confirmation that the certificate has been successfully generated. (The RA or CA may maintain this information for auditing purposes, but should not accept further use of the unique identifier to prove identity.) [Need to address the protection of

the information and, perhaps, provide some guidance on how soon it should be used. The longer it's kept, the greater the risk of exposure.]

- The identity of the public key owner is established at the RA or CA using a previous determination of the public key owner's identity. This is accomplished by "chaining" a new public key certificate request to a previously certified pair (i.e., the signature verification public key). For example, the request for a new public key certificate is signed by the owner of the new public key to be certified. The signing private key used to sign the request **SHOULD** be associated with a verification public key that is certified by the same CA that will certify the new public key. The request **SHOULD** contain the new public key, remaining assurances (key use, parameters, and validation information), and POP.
- The public key, key use, parameters, validation information, and POP are provided to the RA or CA along with a claimed identity. The RA or CA delegates the verification of the public key owner's identity to another trusted process (e.g., an examination of the public key owner's identity by the U.S. postal service when delivering registered mail). Upon receiving a request for certification, the RA or CA generates and sends unique, unpredictable information (e.g., an authenticator or cryptographic key) to the requestor using the trusted process. The trusted process verifies the identity of the requestor prior to delivery of the information provided by the RA or CA. The owner uses this information to prove that the trusted process succeeded, and the RA or CA delivers the certificate to the owner. The information **SHOULD** be destroyed by the key owner as specified in Section 2.3.3.2 upon receiving confirmation that the certificate has been successfully generated. (The RA or CA may maintain this information for auditing purposes, but should not accept further use of the unique identifier to prove identity.)

In cases involving an RA, upon receipt of all information from the requesting entity (i.e., the owner of the new public key), the RA forwards the relevant information to a CA for certification. The CA SHOULD perform any validation or other checks required for the algorithm with which the public key will be used (e.g., public key validation) prior to issuing a certificate. The CA **SHOULD** indicate the checks or validations that have been performed (e.g., in the certificate, or in the CA policy or practices statement). After generation, the certificate is distributed manually or electronically to the RA, the public key owner, or a certificate repository (i.e., a directory) in accordance with the CA's certificate practices statement.

[Discuss how to provide integrity and association protections during distribution.]

2.3.1.5.1.1.3 General Distribution

Public keys may be distributed to entities other than an RA or CA in several ways. Distribution methods include:

- Manual distribution of the public key itself by the owner of the public key (e.g., in a face to face transfer, or by a bonded courier); the assurances listed in Section 2.3.1.5.1.1 **SHOULD** be provided to the recipient prior to the use of the public key.
- Manual (e.g., in a face to face transfer or by receipted mail) or electronic distribution of a public key certificate by the public key owner, the CA, or a certificate repository (i.e., a directory). Assurances listed in Section 2.3.1.5.1.1 that are not provided by the CA (e.g.,

public key validation) **SHOULD** be provided to or performed by the receiver of the public key prior to the use of the key.

- Electronic distribution of a public key (e.g., using a communication protocol with authentication and content integrity) in which the distributed public key is protected by a certified key pair owned by the entity distributing the public key. The assurances listed in Section 2.3.1.5.1.1 **SHOULD** be provided to the receiving entity prior to use of the public key.

2.3.1.5.1.2 Distribution of Ephemeral Public Keys

When used, ephemeral public keys are distributed as part of a secure key establishment protocol. The key establishment process (i.e., the key establishment scheme + the protocol + any associated negotiation) provides a recipient with the assurances listed in Section 2.3.1.5.1.1. The recipient performs public key validation as specified in [FIPS XXX].

2.3.1.5.1.3 Distribution of Centrally Generated Key Pairs

A static key pair may be generated at a central key generation facility within a FIPS 140-2 validated cryptographic module[11] as specified in Section 2.3.1.5.1 for subsequent delivery to the intended owner of the key pair. A signing key pair generated by a central key generation facility for its subscribers will not provide strong non-repudiation for those individual subscribers; therefore, when non-repudiation is required by those subscribers, it is **RECOMMENDED** that the subscribers generate their own signing key pairs. However, if the central key generation facility generates signing key pairs for its own organization and distributes them to members of the organization, then non-repudiation is provided at an organizational level (but not an individual level).

The private key of a key pair generated at a central facility **SHOULD** only be distributed to the intended owner of the key pair. The confidentiality of the centrally generated private key **SHOULD** be protected, and the procedures for distribution **SHOULD** authenticate the recipient's identity as established during user registration (see Section 2.3.1.1).

The key pair may be distributed to the intended owner using an approved manual (e.g., courier, mail or other method specified by the key generation facility) or secure electronic method (e.g., a secure communication protocol). The private key **SHOULD** be distributed in the same manner as a symmetric key (see Section 2.3.1.5.2.2). During the distribution process, each key of the key pair **SHOULD** be provided with the appropriate protections for that key (see Section 2.2.3.1).

When split knowledge procedures are used for the manual distribution of the private key, the key **SHOULD** be split into multiple key components that are the same length as the original key; each key component **SHOULD** provide no knowledge of the original key (e.g., each key component **SHOULD** appear to be generated randomly).

Upon receipt of the key pair, it is **RECOMMENDED** that the owner validate that the public and private keys of the key pair are correctly associated (i.e., check that they work together, for example, checking that the signing private key can be verified by the verification public key),

[11] Alternatively, a key generation may be performed in a facility that is approved for the generation of classified keying material.

and the key pair have the Approved mathematical properties (i.e., public key validation). Validation of key pairs is discussed in [FIPS XXX]. Further certification and/or distribution of the public key **SHOULD NOT** be performed until these checks have been made.

2.3.1.5.2 Generation and Distribution of Symmetric Keys

The symmetric keys used for the encryption and decryption of data or other keys and for the computation of MACs (see Sections 2.2.2.2 and 2.2.2.3) **SHOULD** be determined by an Approved method and **SHOULD** be provided with protection that is consistent with Section 2.2.3.

Symmetric keys **SHOULD** be:

- generated and subsequently distributed either manually (as specified in Section 2.3.1.5.2.2.1), using a public key transport mechanism, or using a previously distributed or agreed upon key encrypting key (see Sections 2.3.1.5.2.1 and 2.3.1.5.2.2, or
- determined using a key agreement scheme (i.e., the generation and distribution are accomplished with one process) (see Section 2.3.1.5.2.3), or
- derived from a master key (see key derivation in Section 2.3.2.3.3).

2.3.1.5.2.1 Key Generation

Symmetric keys determined by key generation methods **SHOULD** be either generated by an Approved random number generation method, created from the previous key during a key update procedure (see Section 2.3.2.3.2), or derived from a master key (see Section 2.3.2.3.3).

When split knowledge procedures are used for the manual distribution of the key (see Section 2.3.1.5.2.2.1), the key **SHOULD** exist as multiple key components that are the same length as the original key; each key component **SHOULD** provide no knowledge of the key (e.g., each key component **SHOULD** appear to be generated randomly and have no identifiable relationship to the key). The keying material may be created and then split into components, or may be created as separate components.

It is **RECOMMENDED** that generation be performed in a FIPS 140-2 validated cryptographic module or in a facility to which access is controlled, and access to the plaintext secret key is prevented[12].

Keys used only for the storage of information (i.e., data or keying material) **SHOULD NOT** be distributed except for backup or to other entities within an organization that may require access to the information protected by the keys. If the keys are used to protect data or keys to be communicated between entities, one of the entities generates the keys and transports (sends) the keys to the other entity(ies) (see Section 2.3.1.5.2.2).

2.3.1.5.2.2 Key Distribution

Keys generated in accordance with Section 2.3.1.5.2.1 as key encrypting keys (used for key wrapping), as master keys to be used for key derivation, or for the protection of communicated

[12] Alternatively, a key generation may be performed in a facility that is approved for the generation of classified keying material.

information are distributed manually (manual key transport) or using an electronic key transport protocol (electronic key transport).

2.3.1.5.2.2.1 Manual Key Distribution

Keys distributed manually (i.e., by other than an electronic key transport protocol) **SHOULD** be protected throughout the distribution process. During manual distribution, the keys **SHOULD** either be encrypted, protected using split knowledge procedures, or be distributed using appropriate physical security procedures. The manual distribution process **SHOULD** assure:

- the authorized distribution of the keys,
- that the entity distributing the keys is trusted by both the entity that generates the keys and the entity(ies) that receives the keys,
- the keys are protected in accordance with Section 2.2.3.1,
- that the keys are received by the authorized recipient.

When distributed in encrypted form, the key **SHOULD** be encrypted by a long or short-term key encrypting key used for key wrapping, or by a key transport public key owned by the intended recipient. These keys **SHOULD** have been distributed to the receiving entity(ies) as specified in this guideline.

When using split knowledge procedures, each key component **SHOULD** be either encrypted or distributed separately via secure channels for delivery to a different individual. Appropriate physical security procedures **SHOULD** be used to protect each key component as sensitive information.

Physical security procedures may be used for all forms of manual key distribution. However, these procedures are particularly critical when the keys are distributed in plaintext form. In addition to the assurances listed above, accountability and auditing of the distribution process (see Sections 3.5 and 3.6) **SHOULD** be used.

2.3.1.5.2.2.2 Electronic Key Distribution/Key Transport

Electronic key distribution or key transport is used to distribute keys via a communication channel (e.g., the Internet or a satellite transmission). Electronic key transport requires the prior distribution of a long or short-term key encrypting key used for key wrapping or a key transport public key as follows:

- A key encrypting key used for key wrapping **SHOULD** be generated and distributed in accordance with Sections 2.3.1.5.2.1 and 2.3.1.5.2.2, or established using a key agreement scheme as defined in Section 2.3.1.5.2.3.
- A key transport public key **SHOULD** be generated and distributed as specified in Section 2.3.1.5.1.

Approved schemes for the transport of keys using the previously established key encrypting key or key transport public key are provided in [FIPS XXX]. The key transport scheme, together with the associated key establishment protocol **SHOULD** provide assurance that:

- the distributed key is not disclosed or modified,
- the keys are protected in accordance with Section 2.2.3.1, and

- the recipient has received the correct key,

2.3.1.5.2.3 Key Agreement

Key agreement is used in a communication environment to establish keying material using public information contributed by all entities in the communication (most commonly, only two entities) without actually sending the keying material. Approved key agreement schemes are provided in [FIPS XXX]. Key agreement requires the availability of key pairs that are used to calculate shared secrets, which are then used to derive secret keys and other keying material (e.g., IVs).

A key agreement scheme uses either static or ephemeral key pairs or both. The key pairs **SHOULD** be generated and distributed as discussed in Section 2.3.1.5.1. The static and ephemeral key pairs and the subsequently derived keying material must be protected as specified in Section 2.2.3.

A key agreement scheme and its associated key establishment protocol **SHOULD** provide the following assurances:

- Each entity in the key establishment process knows the identity of the other entity(ies); this may be achieved by the key agreement scheme or may be achieved by the protocol in which key agreement is performed. The identity of another entity is established (a) by a key agreement scheme when that entity uses a static key pair that has been registered with an infrastructure (e.g., a PKI), or (b) by the use of a key that has been previously established with the receiving entity. In case (b), the previously established key could be used to authenticate that the entity is the same as the entity associated with previous transactions, without ever revealing the entity's real identity. This may be beneficial in special circumstances, such as a whistleblower scenario.
- The keys used in the key agreement scheme are correctly associated with the entities involved in the key establishment process.
- The public keys have been validated in accordance with the method specified in the scheme (see [FIPS XXX]).
- The derived keys are correct (if key confirmation is used).

Keys derived through key agreement and its enabling protocol **SHOULD NOT** be used to protect information until all desired properties have been achieved.

2.3.1.5.3 Generation and Distribution of Other Keying Material

Keys are often generated in conjunction with or are used with other keying material. This other keying material **SHOULD** be protected in accordance with Section 2.2.3.2.

2.3.1.5.3.1 Domain Parameters

Domain parameters are used by some public key algorithms to generate key pairs, to compute digital signatures, or to establish keys. Typically, domain parameters are generated infrequently and used by a community of users for a long period of time. Domain parameters may be distributed in the same manner as the public keys with which they are associated, or they may be

made available at some other accessible site. The domain parameters **SHOULD** be validated prior to use, either by a trusted entity that indicates its "blessing" on the parameters (e.g., a CA), or by the entity's themselves. Domain parameter validation is addressed in [FIPS XXX]. Domain parameter validation should be addressed in a CA's certificate practices statement or an organization's security plan.

2.3.1.5.3.2 Initialization Vectors

Initialization vectors (IVs) are used by symmetric algorithms in several modes of operation for encryption and decryption, or for authentication. The criteria for the generation and use of IVs is provided in [MODES]; IVs **SHOULD** be protected as specified in Section 2.2.3.2. When distributed, IVs may be distributed in the same manner as their associated keys, or may be distributed with the information that uses the IVs as part of the encryption or authentication mechanism.

2.3.1.5.3.3 Shared Secrets

Shared secrets are computed during a key agreement process and are subsequently used to derive keying material. Shared secrets are generated as specified by the appropriate key agreement scheme (see [FIPS XXX]), but **SHOULD NOT** be distributed.

2.3.1.5.3.4 Secret and Public Seeds

Seeds are used to initialize a pseudorandom number generator (PRNG). The criteria for the selection of a seed is provided in the specification of an Approved PRNG. Secret seeds **SHOULD NOT** be distributed; public seeds may be distributed using a method that will protect the public seed as specified in Section 2.2.3.2.

2.3.1.5.3.5 Intermediate Results

Intermediate results occur during computation using cryptographic algorithms. These results **SHOULD NOT** be distributed.

2.3.1.6 Key Registration

Key registration results in the binding of keying material to information or attributes associated with a particular entity. This information typically includes the identity of the entity associated with the keying material and the intended use of the keying material (e.g., signing key, data encryption key, etc.). Additional information may include authorization information or specify the level of trust. The binding provides assurance to the community at large that the keying material is used by the correct entity in the correct application. The binding is often cryptographic, which creates a strong association between the keying material and the entity. A trusted third party performs the binding. Examples of a trusted third party include a Kerberos realm server or a PKI certification authority (CA).

When a Kerberos realm server performs the binding, a symmetric key is stored on the server with the corresponding attributes, In this case, the registered keying material is maintained in confidential storage (i.e., the keys are provided with confidentiality protection).

When a CA performs the binding, the public key and associated attributes are placed in a public key certificate, which is digitally signed by the CA. In this case, the registered key material may be publicly available.

When a CA provides a certificate for signature verification public keys, the keys **SHOULD** be verified to ensure that they are indeed associated with the private signing key known by the purported owner of the public key. This is commonly known as proof-of-possession (POP). As a general rule, the POP **SHOULD** be accomplished using the keys for their intended function. For example, the POP of a key transport key should be done using a key transport function, not a digital signature function.

2.3.2 Operational Phase

Keying material used during the cryptoperiod of a key is often stored for access as needed. During storage, the keying material **SHOULD** be protected as specified in Section 2.2.3.2.2. During normal use, the keying material is stored either on the device or module that uses that material, or on a readily accessible storage media. When the keying material is required for operational use, the keying material is acquired from immediately accessible storage when not present in active memory within the device or module.

To provide continuity of operations when the keying material becomes unavailable for use from normal operational storage during its cryptoperiod (e.g., because the material is lost or corrupted), keying material may need to be recoverable. If an analysis of system operations indicates that the keying material needs to be recoverable, then the keying material **SHOULD** be backed up (see Section 2.3.2.2.1), or the system **SHOULD** be designed to allow re-creation (e.g., rederivation) of the keying material. Acquiring the keying material from backup or by re-derivation is commonly known as key recovery (see Section 2.3.2.2.2).

At the end of a key's cryptoperiod, a new key needs to be available to replace the old key if operations are to be continued. This can be accomplished by rekeying (see Section 2.3.2.2.3), key update (see Section 2.3.2.2.4) or key derivation (see Section 2.3.2.2.5). A key **SHOULD** be destroyed in accordance with Section 2.3.3.3 as soon as that key is no longer needed in order to reduce the risk of exposure.

2.3.2.1 Normal Operational Storage

The objective of the key management lifecycle is to facilitate the operational availability of keying material for standard cryptographic purposes. Usually, a key remains operational until the end of the key's cryptoperiod (i.e., the expiration date). During normal operational use, keying material is available either in the device or module (e.g., in RAM) or in an immediately accessible storage media (e.g., on a local hard disk).

2.3.2.1.1 Device or Module Storage

Keying material may be stored in the device or module that adds, checks or removes the cryptographic protection on information. The storage of the keying material **SHOULD** be consistent with Section 2.2.3.2.2, as well as with FIPS 140-2.

2.3.2.1.2 Immediately Accessible Storage Media

Keying material may need to be stored for normal cryptographic operations on an immediately accessible storage media (e.g., a local hard drive) during the cryptoperiod of the key. The storage requirements of Section 2.2.3.2.2 apply to this keying material.

2.3.2.2 Continuity of Operations

Keying material can become lost or unusable due to hardware damage, corruption or loss of program or data files, or system policy or configuration changes. In order to maintain the continuity of operations, it is often necessary for users and/or administrators to be able to recover keying materials from back-up storage. However, if operations can be continued without the backup of keying material, or the keying material can be recovered or reconstructed without being saved, it may be preferable not to save the keying material in order to lessen the possibility of a compromise of the keying material or other cryptographic related information.

The compromise of keying material affects continuity of operations (see Section 3.4). When keying material is compromised, continuity of operations requires the establishment of entirely new key material (see Section 2.3.1.5), following an assessment of what keying material is affected and must be replaced.

2.3.2.2.1 Backup Storage

The backup of keying material on an independent, secure storage media provides a source for key recovery (see Section 2.3.2.2.2). Backup storage is used to store copies of information that is also currently available in normal operational storage during a key's cryptoperiod (i.e., in the cryptographic device or module, or on an immediately accessible storage media - see Section 2.3.2.1.2). Not all keys need be backed up. The storage requirements of Section 2.2.3.2.2 apply to keying material that is backed up. Tables 4-6 provide guidance about the backup of each type of keying material and other related information. An "OK" indicates that storage is permissible, but not necessarily required. It is **RECOMMENDED** that the final determination for backup be made based on the application in which the keying material is used. A detailed discussion about each type of key and other cryptographic information is provided in Appendix B.3.

Keying material maintained in backup **SHOULD** remain in storage for at least as long as the same keying material is maintained in storage for normal operational use (see Section 2.3.2.1). When no longer needed for normal operational use, it is **RECOMMENDED** that the keying material and other related information be removed from backup storage. When removed from backup storage, all traces of the information **SHOULD** be destroyed in accordance with Section 2.3.3.3.

A discussion of backup and recovery is provided in [ITL Bulletin].

Table 4. Backup of keys.

Type of Key	Backup?
Signing private key	No (in general); non-repudiation would be in question. However, it may be warranted in some cases - a CA's signing private key, for example.
Signature verification public key	OK; its presence in a public-key certificate that is available elsewhere may be sufficient.
Secret authentication key	OK
Authentication private key	OK, if required by an application.
Authentication public key	OK; its presence in a public-key certificate that is available

	elsewhere may be sufficient.
Long-term data encryption key	OK
Short-term data encryption key	OK
RNG key	Not necessary and may not be desirable, depending on the application.
Long-term key encrypting key used for key wrapping	OK
Short-term key encrypting key used for key wrapping	OK
Master key used for key derivation	OK
Key derived from a Master Key	Depends on the use of the derived key, but backup may not be needed if the master key is backed up.
Key transport private key	OK
Key transport public key	OK; presence in a public-key certificate available elsewhere may be sufficient.
Secret key agreement key	OK
Static key agreement private key	No, unless needed for reconstruction during key recovery. However, when ephemeral information (e.g., an ephemeral key agreement private key) is used in a key agreement scheme, knowledge of the static key agreement private key and any public keys will not be sufficient.
Static key agreement public key	OK; its presence in a public-key certificate that is available elsewhere may be sufficient.
Ephemeral key agreement private key	No
Ephemeral key agreement public key	No, unless needed for reconstruction during key recovery
Secret authorization key	OK
Authorization private key	OK
Authorization public key	OK; its presence in a public-key certificate that is available elsewhere **MAY** be sufficient.

Table 5. Backup of other keying material.

Type of Keying Material	Backup?
Domain parameters	OK
Initialization vector	OK, if necessary

Shared secret	No, unless needed for reconstruction during key recovery
Secret seed	No
Public seed	OK, if required for the validation of domain parameters
Nonce	OK
Intermediate results	No

Table 6. Backup of other related information.

Type of Information	Backup?
Hash value	OK
Encrypted data	OK
MAC	OK
Digital signature	OK
Key control information (e.g., IDs, purpose, etc.)	OK
Random number	OK
Passwords	OK
Audit information	OK

2.3.2.2.2 Key Recovery

Keying material that is in active memory or stored in normal operational storage may sometimes be lost or corrupted (e.g., from a system crash or power fluctuation). Some of the keying material is needed to continue operations and cannot easily be replaced. An assessment needs to be made of which keying material needs to be preserved for possible recovery at a later time.

The decision as to whether key recovery is required **SHOULD** be made on a case by case basis. The decision **SHOULD** be based on:

- the type of key (e.g., signing private key, long-term data encryption key),
- the application in which the key will be used (e.g., interactive communications, file storage),
- whether the key is "owned" by the local entity (e.g., a private key) or by another entity (e.g., the other entity's public key) or is shared (e.g., a data encyption key shared by two entities),
- the role of the entity in a communication (e.g., sender of receiver), and

- the algorithm or computation in which the key will be used (e.g., does the entity have the necessary to information to perform a given computation if the key were to be recovered)[13].

The factors involved in a decision for or against key recovery **SHOULD** be carefully assessed. The trade-offs are concerned with continuity of operations versus the risk of possibly exposing the keying material and the information it protects if control of the keying material is lost. If it is determined that a key needs to be recoverable, and the key is still active (i.e., the cryptoperiod of the key has not expired), then the key **SHOULD** be replaced as soon as possible in order to limit its exposure (see Section 2.3.2.3).

Issues associated with key recovery and discussions about whether or not different types of cryptographic material need to be recoverable are provided in Appendix B.

2.3.2.3 Key Replacement

Prior to or at the end of a key's cryptoperiod, the key needs to be replaced by a new key if a cryptographic capability is to continue. A key may also need to be replaced for other reasons, such as key compromise or the unavailability of the key's owner to perform the cryptographic function. A key is replaced by either rekeying, key update or key derivation.

2.3.2.3.1 Rekeying

If the new key is generated in a manner that is entirely independent of the "value" of the old key, the process is known as rekeying. This replacement **SHOULD** be accomplished using one of the key establishment methods discussed in Section 2.3.1.5.

2.3.2.3.2 Key Update

If the "value" of the new key is dependent on the value of the old key, the process is known as key update (i.e., the current key is modified to create a new key). This **SHOULD** be accomplished by applying a one-way function to the old key and other data. Unlike rekeying, key update does not require the exchange of any new information between the entities that previously shared the old key. For example, the two entities may just agree to update their keys on the first day of each month. Since a one-way function was used in the update process, key update has the property that previous keys are protected in the event that a key is compromised. However, future keys are not protected. After a limited number of updates, new keying material **SHOULD** be established by employing a fresh rekey operation (see Section 2.3.2.3.1).

[What additional guidance can be provided for the number of updates before a rekeying is necessary and the evaluation of update procedures? What other guidance is needed?]

2.3.2.3.3 Key Derivation

Secret and private cryptographic keys may be derived from other secret values. Four of the cases are discussed below:

[We need to determine which of these to include; currently, none are defined in a FIPS.]

---

[13] This could be the case when performing a key establishment process for some key establishment schemes (see FIPS XXX).

1. Two parties derive common keys from a common shared secret. This approach is used in the key agreement techniques specified in [FIPS XXX]. The security of this process is dependent on the security of the shared secret and the specific key derivation function used. If one knows the shared secret, the derived keys may be determined. The key derivation function specified in [FIPS XXX] **SHOULD** be used for this purpose. [Do we want to keep this?]
2. Individual entity keys are derived from a master key. This is often accomplished by using the master key, entity ID, and other known information as input to a function that generates the entity keys. The security of this process depends upon the security of the master key and the key derivation function. If one of the entities knows the master key, the other entity keys may all be generated. Keys derived from a master key **SHOULD** be used for authentication purposes, only. Thus, these derived keys may be used to authenticate the identity of entities corresponding to these keys, but may not be used for general data encryption.
3. The individual entity key is derived from a master key and the entity password. These secret values are input to the key derivation function along with other known information. The security of a derived entity key is dependent upon the security of the master key, the security of the password, and the strength of the key derivation process. Keys derived in this manner **SHOULD** be used for authentication purposes only and not for general encryption.
4. The individual entity key is derived from the entity password[14]. This is done by using a password, entity ID, and other known information as input to the key derivation function. This technique differs from the previous method in that no master key is used. Therefore, the security of the process depends upon the security of the password and the key derivation process. If the entity password is known or can be guessed, then the corresponding derived entity key may be generated. Keys derived from a password **SHOULD** be used for authentication purposes only and not for general encryption.

[Other key derivation techniques exist. We may decide to include them in this list.]

2.3.3 Post-Operational Phase

During the post-operational phase, keying material is no longer in operational use, but access to the keying material is still possible.

2.3.3.1 Archive Storage and Key Recovery

If the keying material needs to be recoverable after the end of its cryptoperiod, the keying material **SHOULD** be either archived, or the system **SHOULD** be designed to allow re-creation (i.e., rederivation) of the keying material. Acquiring the keying material from archive storage, or by re-derivation is commonly known as key recovery.

A key management archive is a repository containing keying material and other related information of historical interest. Not all keying material needs to be archived. It is

[14] This technique is used in RSA PKCS #5.

RECOMMENDED that an organization's security plan indicate the types of information that is to be archived.

While in storage, archived information may be either static (i.e., never changing) or may need to be re-encrypted under a new archive encryption key. Archived data **SHOULD** be stored separately from operational data, and it is **RECOMMENDED** that multiple copies of archived cryptographic information be provided in physically separate locations (i.e., it is **RECOMMENDED** that the key management archive be backed up). For critical information encrypted under archived keys, it may be necessary to back up archive keys and to store multiple copies of archived keys in separate locations.

When archived, it is **RECOMMENDED** that keying material be archived prior to the end of the validity period (i.e., cryptoperiod, reliance period) of the key. When no longer required, the keying material **SHOULD** be destroyed in accordance with Section 2.3.3.3.

Archived cryptographic information requires protection in accordance with Section 2.2.3.2.2. Confidentiality is provided by an archive encryption key (one or more encryption keys that are used exclusively for the encryption of archived information), by another key that has been archived, or by a key that may be derived from an archived key. When encrypted by the archive encryption key, the encrypted keying material **SHOULD** be re-encrypted by any new archive encryption key at the end of the cryptoperiod of the old archive encryption key. When the keying material is re-encrypted, integrity values on that keying material **SHOULD** be recomputed.

Likewise, integrity protection is provided by an archive integrity key (one or more authentication or digital signature keys that are used exclusively for the archive) or by another key that has been archived. At the end of the cryptoperiod of the archive integrity key, new integrity values **SHOULD** be computed on the archived information on which the old archive integrity key was applied.

The archive keys may be either symmetric (secret) keys, or key pairs. The keys used for confidentiality and integrity **SHOULD** be different, and **SHOULD** be protected in the same manner as their key type (see Sections 2.2.3.1.1 and 2.2.3.2.2).

The key archiving technique used **SHOULD** be appropriate to the storage method used for the encrypted information protected under archived keys. [Are the archive encryption or integrity keys themselves archived?] For example, the differences between the organization of random access data files and database management system (DBMS) files are such that a method used to store keys under which random access files are protected and to associate the keys with the protected data would likely be both inappropriate and inadequate for DBMS applications. (Assume that the elements stored in a database are created at different times and protected under different keys. The key information would then need to be associated with data elements rather than data files.)

Tables 7-9 indicate the desirability of archiving the various key types. An "OK" in column 2 (Archive?) indicates that archival is permissible, but not necessarily required. Column 3 (Retention period) indicates the minimum time that the key **SHOULD** be retained in the archive. Column 4 (Alterable?) indicates conditions under which the keying material is altered.

Additional advice on the storage of keying material in archive storage is provided in Appendix B.3.

Table 7: Archive of keying material and other related information.

Type of Key	Archive?	Retention period (minimum)
Signing private key	No	
Signature verification public key	OK	Until no longer required to verify data signed with the assoc. private key
Secret authentication key	OK	Until no longer needed to authenticate data.
Authentication private key	No	
Authentication public key	OK	Until no longer required to verify the authenticity of data that was authenticated with the assoc. private key
Long-term data encryption key	OK	Until no longer needed to decrypt data encrypted by this key
Short-term data encryption key	OK	Until no longer needed to decrypt data encrypted by this key
RNG key	No	
Long-term key encrypting key used for key wrapping	OK	Until no longer needed to decrypt keys encrypted by this key
Short-term key encrypting key used for key wrapping	OK	
Master key used for key derivation	OK, if needed to derive other keys for archived data	Until no longer needed to derive other keys
Keys derived from a Master Key	Depends on use	Depends on use
Key transport private key	OK	Until no longer needed to decrypt keys encrypted by this key
Key transport public key	No	
Secret key agreement key	No	
Static key agreement private key	No, unless needed to reconstruct keying material	Until no longer needed to reconstruct keying material

Static key agreement public key	No, unless needed to reconstruct keying material	Until no longer needed to reconstruct keying material
Ephemeral key agreement private key	No	
Ephemeral key agreement public key	No, unless needed to reconstruct keying material	Until no longer needed to reconstruct keying material
Secret authorization key	No	
Authorization private key	No	
Authorization public key	No	

Table 8: Archive of Other Keying Material.

Type of Key	Archive?	Retention period (minimum)
Domain parameters	OK	Until all keying material, signatures and signed data using the domain parameters are removed from the archive
Initialization vector	OK; normally stored with the protected information	Until no longer needed to process the protected data
Shared secret	No, unless needed to validate or reconstruct derived keying material for archived information	Until no longer needed to validate or reconstruct derived keying material for archived information.
Secret seed	No	
Public seed	OK	Until no longer needed to process generated data
Nonce	OK	Until no longer needed to process data using the nonce
Intermediate result	No	

Table 9: Archive of Other Information Requiring Protection

Type of Key	Archive?	Retention period (minimum)
Hash value	OK	Until the data assoc. with the hash value is no longer needed.
Encrypted data	OK	Until the data is no longer required.
MAC	OK	Until the authenticity of data authenticated by the key no longer needs to be checked.
Digital signature	OK	Until the signature on data signed with the assoc. private key no longer needs to be verified
Key control information (e.g., IDs, purpose)	OK	Until the associated key is removed from the archive
Random number	No, unless needed for reconstruction	Until no longer needed for reconstruction
Password	No	
Audit information	OK	Until no longer needed

After the end of a key's cryptoperiod, keying material may be recovered from archival storage, providing that the keying material has been archived. Alternatively, the keying material may be re-created (i.e., rederived), if the key management system has been appropriately designed.

Key recovery of archived keying material may be required to remove (e.g., decrypt) or check (e.g., verify a digital signature or a MAC) the cryptographic protections on archived data. The key recovery process results in retrieving the desired keying material from archive storage and placing it in active memory or normal operational storage in order to perform the required cryptographic operation. Immediately after completing this operation, the keying material **SHOULD** be erased from the active memory and normal operational storage. Further advice on key recovery issues is provided in Appendix B.

2.3.3.2 User De-registration

When an entity ceases to be a member of a security domain, the entity must be de-registered. De-registration is intended to prevent other entities from relying on or using the de-registered entity's keying material.

All records of the entity and the entity's associations **SHOULD** be marked to indicate the entity is no longer a member of the security domain, but the records **SHOULD NOT** be deleted. To reduce confusion and unavoidable human errors, identification information associated with the de-registered entity **SHOULD NOT** be re-used (at least for a period of time). For example, if a "John Wilson" retires and is de-registered on Friday, the identification information assigned to his son "John Wilson", who is hired the following Monday, **SHOULD** be different.

2.3.3.3 Key De-registration

Registered keying material may be associated with the identity of a key owner, owner attributes (e.g., email address), or role or authorization information. When the keying material is no longer needed, or the associated information becomes invalid, the keying material **SHOULD** be de-registered (i.e., all records of the keying material and its associations **SHOULD** be marked to indicate that the key is no longer in use).

When a cryptographic key is compromised, that key and any associated keying material **SHOULD** be de-registered. For example, when a private key is compromised, the corresponding public key certificate **SHOULD** be revoked. Certificate revocation because of a key compromise advertises that the binding between the owner and the key is no longer to be trusted. In a PKI, key de-registration is commonly achieved by including the certificate in a list of revoked certificates (i.e., a CRL). Where the PKI uses online status mechanisms (e.g., the Online Certificate Status Protocol [OCSP]), de-registration is achieved by informing the appropriate certificate status server(s).

Keying material **SHOULD** be de-registered when the attributes associated with an entity are modified. For example, if an entity's email address is associated with a public key, and the entity's address changes, the keying material **SHOULD** be de-registered to indicate that the associated attributes have become invalid. Unlike the case of key compromise, the entity could safely re-register the public key after modifying the entity's attributes through the user registration process (see Section 2.3.1.1).

2.3.3.4 Key Destruction

All copies of the private or secret key **SHOULD** be destroyed as soon as no longer required (e.g., for archival or reconstruction activity) in order to minimize the risk of a compromise. Any media on which unencrypted keying material requiring confidentiality protection was stored **SHOULD** be erased in a manner that removes all traces of the keying material so that it cannot be recovered by either physical or electronic means[15]. Public keys may be retained or destroyed, if desired; retention does not introduce a security problem.

2.3.3.5 Key Revocation

It is sometimes necessary to remove keying material from use prior to the end of its normal cryptoperiod for reasons that include key compromise, removal of an entity from an organization, etc. This process is known as key revocation and is used to explicitly revoke a secret key or the public key of a key pair, although the private key associated with public key is also revoked.

Key revocation may be accomplished using a notification indicating that the continued use of the keying material is no longer **RECOMMENDED**. The notification could be provided by actively sending the notification to all entities that might be using the revoked keying material, or by allowing the entities to request the status of the keying material (i.e, a “push” or a “pull” of the status information). It is **RECOMMENDED** that the notification include a complete

[15] A simple deletion of the keying material might not completely obliterate the information. Erasing the information might require overwriting that information with other non-related information, such as random bits, or all zero or one bits.

identification of the keying material, the date and time of revocation and the reason for revocation, when appropriate (e.g., key compromised). Based on the revocation information provided, other entities could make a determination of how they would treat information protected by the revoked keying material.

For example, if a signature verification public key is revoked because an entity left an organization, it may be appropriate to honor all signatures created prior to the revocation date. If a signing private key is compromised, an assessment needs to be made as to whether or not information signed prior to the revocation would be considered as valid.

As another example, a secret key used to generate MACs may be revoked so that it is not used to generate MACs on new information. However, the key may be retained so that archived documents can be verified.

The details for key revocation will reflect the lifecycle for each particular key. If a key is used in a pair-wise situation (e.g., two entities communicating in a secure session) the entity revoking the key simply informs the other entity. If the key has been registered with an infrastructure, the entity revoking the key cannot directly inform the other entities that may rely upon that key. Instead, the entity revoking the key informs the infrastructure that the key should be revoked (e.g., using a certificate revocation request). The infrastructure responds by de-registering the key material (see 2.3.3.3).

2.3.4 Obsolete/Destroyed Phase

The keying material is no longer available. All records of its existence may have been deleted. However, some organizations may require the retention of key management records for audit purposes. For example, if a copy of an ostensibly destroyed key is found in an uncontrolled environment, records of the identity of the key and its use may be helpful in determining what was protected under the key, the probability that the information or process was compromised, and how to recover from any assumed compromise.

3 GENERAL KEY MANAGEMENT GUIDANCE

This section is intended to provide general guidance concerning key usage, cryptoperiod length, domain parameter validation and public key validation, key compromise, accountability, audit and key management system survivability. In addition, guidance is provided for selecting appropriate algorithms and key sizes and using the key establishment techniques specified in [FIPS XXX].

3. 1 Key Usage

It is a general precept of secure key management that a single key **SHOULD** be used for only one purpose (e.g., encryption, authentication, key wrapping, random number generation, or digital signatures). There are several reasons for this:

- Each use of a key exposes it to attack in some way, and using the key for different cryptographic processes exposes the key to different attacks.
- Limiting the use of a key limits the damage that could be done if the key is compromised.
- Some uses of keys interfere with each other. For example, consider a key pair used for both key transport and digital signatures. A private key that is used to decrypt a data encryption key that is, in turn, used for the decryption of an encrypted file needs to be retained for use as long as the data in the encrypted file needs to be accessed. The retention period may be longer than the cryptoperiod of the associated public key that was used to encrypt the data encryption key. On the other hand, a public key used to validate digital signatures needs to be retained as long as the digital signature on any information signed by the associated signing private key may need to be verified. However, the associated signing private key **SHOULD** be destroyed at the expiration of its validity period to prevent its compromise. In this example, the longevity requirements for key transport key pairs and digital signature key pairs contradict each other.

This principle does not preclude using a single key in cases where the same process can provide multiple services. This is the case, for example, when a digital signature provides non-repudiation, authentication and integrity protection using a single digital signature, or when a single secret key can be used to encrypt and authenticate data in a single cryptographic operation (e.g., using an authenticated encryption operation, as opposed to separate encryption and authentication operations). Also refer to Section 2.1.7.

3.2 Cryptoperiods

[There was a comment that the revocation strategy tends to drive the cryptoperiod. May need to deal with this.]

A cryptoperiod is the time span during which a specific key is authorized for use by legitimate entities, or the keys for a given system will remain in effect. A suitably defined cryptoperiod:

1. limits the amount of information protected by a given key that is available for cryptanalysis,
2. limits the amount of exposure if a single key is compromised,

3. limits the use of a particular algorithm to its estimated effective lifetime, and
4. limits the time available for computationally intensive cryptanalytic attacks (in applications where long-term key protection is not required).

Trade-offs associated with the determination of cryptoperiods involve the risk and consequences of exposure.

Among the factors affecting the risk of exposure are:

- the strength of the cryptographic mechanisms (e.g., the algorithm, key length, mode of operation),
- the embodiment of the mechanisms (e.g., FIPS 140-2 Level 4 implementation, or software implementation on a Microsoft Windows machine),
- the operating environment (e.g., secure limited access facility, open office environment, or publicly accessible terminal),
- the volume of information flow or the number of transactions,
- the security life of the data,
- the security function (e.g., data encryption, digital signature, key production or derivation, key protection),
- the rekeying method (e.g., keyboard entry, rekeying using a key loading device where humans have no direct access to key information, remote rekeying within a PKI),
- the key update or key derivation process,
- the number of nodes in a network that share a common key, and
- the threat to the information (e.g., who the information is protected from, and what are their perceived technical capabilities and financial resources to mount an attack).

In some cases, increased risk might suggest shorter cryptoperiods, while, in other cases, increased risk might suggest a need for longer cryptoperiods. For example, some cryptographic algorithms might be more vulnerable to cryptanalysis if the adversary has access to large volumes of stereotyped data that is encrypted under the same key. Particularly where a secret key is shared among several entities, it may be prudent to employ short cryptoperiods for such algorithms. On the other hand, where manual key distribution methods are subject to human error and frailty, more frequent key changes might actually increase the risk of exposure. In these cases, especially when very strong cryptography is employed, it may be more prudent to have fewer, well-controlled manual key distributions rather than more frequent, poorly controlled manual key distributions. In some cases, the cost associated with changing keys is prohibitive (e.g., decryption and subsequent re-encryption of very large databases, physical replacement of a very large number of keys). In such cases, the use of cryptography that is strong enough to support longer cryptoperiods may be justified, even where routine data and/or retrieval operations result in significant processing overhead[16].

[16] One must always keep in mind the potentially increased requirements for the physical/logical protection of keys as cryptoperiods increase.

The consequences of exposure are measured by the sensitivity of the information, the criticality of the processes protected by the cryptography, and the cost of recovery from the compromise of the information or processes. Sensitivity refers to the lifespan of the information being protected (e.g., 10 minutes, 10 days or 10 years) and the potential consequences of a loss of protection for that information (e.g., the disclosure of the information to unauthorized entities). In general, as the sensitivity of the information or the criticality of the processes protected by cryptography increase, the length of the associated cryptoperiods **SHOULD** decrease in order to limit the damage that might result from each compromise. This is, of course, subject to the caveat regarding the security and integrity of the rekeying , key update or key derivation process (see Section 2.3.2.3). Particularly where denial of service is the paramount concern, and there is a significant potential for error in the rekeying, key update or key derivation process, short cryptoperiods may be counterproductive.

In some cases, a key may have different cryptoperiods for its use in originating cryptographic protection and the subsequent processing of protected information by a recipient.

- For key pairs, each key of the pair has its own cryptoperiod; each key is used by an "originator" to apply cryptographic protection (e.g., create a digital signature) or by a "recipient" to subsequently process the protected information (e.g., verify a digital signature), but not both. Examples of this distinction include:
  - The cryptoperiod of key transport private key is longer than the cryptoperiod of the associated public key (i.e., the key transport public key). The public key is used to encrypt keying material for a fixed period of time, which may be indicated by the expiration date on a public has been recovered (i.e., decrypted) or destroyed.
  - The cryptoperiod of an authentication private key that is used to sign challenge information is basically the same as the cryptoperiod of the associated public key (i.e., the authentication public key). That is, when the private key will not be used to sign challenges, the public key is no longer needed.
  - If a signing private key is used to generate digital signatures as a proof-of-origin in the future, the cryptoperiod of the private key is significantly shorter than the cryptoperiod of the associated verification public key. In this case, the private key is usually intended for use for a fixed period of time, which may be indicated by the expiration date on the corresponding public key certificate. After the expiration of the certificate, the key owner should destroy the private key. The public key must be available for the longest period of time that the signature may need to be verified. In this case, the effective cryptoperiod for the public key may be extended by supplementing the strength of the mechanisms (e.g., the digital signature on the certificate) by submitting the certificate and signed object to a trusted electronic archive, or by obtaining a cryptographic timestamp.
- For symmetric keys, a single key is used for both applying the protection (e.g., encrypting or computing a MAC) and processing the protected information (e.g., decrypting the encrypted information or verifying a MAC). The cryptoperiod of the key consists of an "originator usage period" and a "recipient usage period". The originator usage period pertains to the time during which the cryptographic protection is originally applied (e.g., the information is originally encrypted, or the MAC is originally computed); the key **SHOULD NOT** be used to apply this protection at a later time (e.g., the key **SHOULD**

NOT be used for encrypting information or computing the original MAC after the end of the originator usage period). The recipient usage period pertains to the time during which the key can be used for the subsequent processing of the protected information (e.g., decrypting the encrypted information or verifying the MAC). The usage periods begin at the beginning of the cryptoperiod; the recipient usage period may extend beyond the originator usage period; the recipient usage period ends at the end of the cryptoperiod. In many cases, the originator and recipient usage periods are the same. See Figure 1.

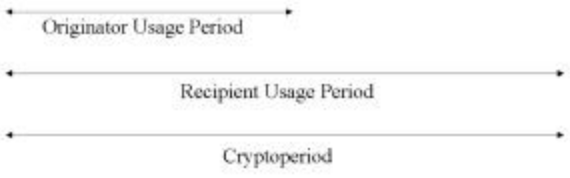

Figure 1: Symmetric Key Cryptoperiod

Examples of the use of the usage periods include:

- When a symmetric key is used for securing communications, the difference between the originator's application of protection and the recipient's processing is negligible. In this case, the key is authorized for either purpose during the entire cryptoperiod, i.e., the originator usage period and the recipient usage period are the same.
- When a symmetric key is used to protect stored information, the originator usage period (when the originator applies cryptographic protection to stored information) may end much earlier than the recipient usage period (when the stored information is processed). In this case, the cryptoperiod begins at the initial time authorized for application of protection with the key, and ends with the latest time authorized for processing using that key. In general, the recipient usage period for stored information will continue beyond the originator usage period, so that the stored information may be authenticated or decrypted.

[May need text about the cryptoperiods keys used for communication vs. keys used for storage]

[The intent is to collapse the following list into classes of keys to eliminate the repetitiveness. Suggestions for doing so would be appreciated.]

To facilitate discussions, the following discussions are provided for the different key types.

1. *Signing private key*: The cryptoperiod of a signing private key may, in general, shorter than the cryptoperiod of the corresponding signature verification public key. For further advice, see Appendix C. [Need to provide guidance on the specific length of the cryptoperiod - depends on the amount of information to be signed - maximum about 1-2 years or XXX uses?]
2. *Signature verification public key*: The cryptoperiod of a signature verification public key may, in general, longer than the cryptoperiod of the corresponding signing private key. The cryptoperiod is, in effect, the period during which any signature computed using the associated signing private key needs to be verified. A long cryptoperiod for the signature verification public key poses no security concern. For further advice, see Appendix C.
3. *Secret authentication key*: The cryptoperiod of a secret authentication key depends on the sensitivity of the type of information it protects. For very sensitive information (e.g., information that one would not like to be compromised (unprotected) if the key used to

protect other information were compromised), the authentication key may need to be unique to the protected information. Otherwise, suitable cryptoperiods may extend beyond a single use, up to a period of 1-2 years. The originator usage period of a secret authentication key applies to the use of that key in applying the original cryptographic protection for the information (e.g., computing the MAC to be associated with the authenticated information); new MACs **SHOULD NOT** be computed on information after the end of the originator usage period. However, the key **SHOULD** be available to verify the MAC on the protected data for as long as the protection is required, i.e., the recipient usage period of the key, when used only for verification purposes, is the lifetime of the protected information. [Note: This statement and similar statements for symmetric keys may change (these statements are highlighted in this section). Cryptoperiods are normally pre-determined and may not be adequate for the lifetime of the protected data. This problem needs to be addressed.]

4. A*uthentication private key*: An authentication private key may be used multiple times. Its associated public key could be certified, for example, by a Certificate Authority. In most cases, the cryptoperiod of the authentication private key is the same as the cryptoperiod of the associated public key. An appropriate cryptoperiod for the key would be 1-2 years, depending on its use.

5. *Authentication public key*: In most cases, the cryptoperiod of an authentication public key is the same as the cryptoperiod of the associated authentication private key. The cryptoperiod is, in effect, the period during which the authentication of information protected by the associated authentication private key needs to be verified.

6. *Long-term data encryption key*: A long-term data encryption key is used multiple times over an extended period of time. A data encryption key that is used to encrypt large volumes of information over a short period of time (e.g., for a link encryption) **SHOULD** have a relatively short cryptoperiod (e.g., a day or a week). An encryption key used to encrypt less information could have a longer cryptoperiod (up to one month?). The originator usage period of a long-term data encryption key applies to the use of that key in applying the original cryptographic protection for information (i.e., encrypting the information). During the originator usage period, information may be encrypted by the data encryption key; the key **SHOULD NOT** be used for encrypting information beyond this period. However, the key **SHOULD** be available to decrypt the encrypted information for as long as the encrypted information exists, i.e., the recipient usage period of the key, when used only for decryption, is the lifetime of the encrypted information.

7. *Short-term data encryption key*: The cryptoperiod of a short-term data encryption key is very short, e.g., a single message or a communication session. The originator usage period of a short-term data encryption key applies to the use of that key in applying the original cryptographic protection for information (i.e., encrypting the information). During the originator usage period, information may be encrypted by the data encryption key; the key **SHOULD NOT** be used for encrypting information beyond this period. However, the key **SHOULD** be available to decrypt the encrypted information for as long as the encrypted information exists, i.e., the recipient usage period of the key, when used only for decryption, is the lifetime of the encrypted information.

8. *RNG key*: The cryptoperiod of a key used to create random numbers depends on the amount of its use, though it may be used for an extended period of time. [Suitable cryptoperiods might be a month or a year or two. Hopefully the RNG standard under development (ANSI X9.82) will allow us to provide better advice.]

9. *Long-term key encrypting key used for key wrapping*: This key is used multiple times over an extended period of time. A key of this type that is used to encrypt large numbers of keys over a short period of time **SHOULD** have a relatively short cryptoperiod (e.g., a day or a week). If a small number of keys are encrypted, the cryptoperiod of the key encrypting key could be longer (up to a month?). The originator usage period of a long-term key encrypting key applies to the use of that key in providing the original protection for information (i.e., encrypting the key that is to remain secret); keys **SHOULD NOT** be encrypted using the key encrypting key after the end of the originator usage period. However, the key encrypting key **SHOULD** to be available to decrypt the encrypted key for as long as the encrypted key exists in its encrypted form, i.e., the recipient usage period of the key encrypting key, when used only for decryption, is the lifetime of the encrypted key.

10. *Short-term key encrypting key used for key wrapping*: The cryptoperiod of a short-term key encryption key is very short, since it is used to wrap a very few keys. An appropriate cryptoperiod would be a single communication session. It is assumed that the key as encrypted by the key encrypting key will not be retained in its encrypted form, so the cryptoperiod of the key encrypting key as used for encryption (the originator usage period) is the same as that same key used for decryption (the recipient usage period).

11. *Master key used for key derivation*: A master key is used multiple times to derive other keys. Therefore, a suitable cryptoperiod depends on the considerations provided earlier in this section. [Need to provide more guidance.]

12. *Keys derived from a master key*: The cryptoperiod of a key derived from a master key could be relatively short, e.g., a single use or a communication session or transaction. Alternatively, keys derived from the same master key could be used for entirely different purposes (e.g., one key for encryption, another key for HMAC computations). The originator usage period of a derived key applies to the use of that key in applying the original cryptographic protection for information (e.g., encrypting the information). The key may need to be available to process the protected information for as long as the protected information exists (i.e., the recipient usage period of the derived key, when used only for the subsequent processing of the protected information, is the lifetime of the protected information). For example, if a key is derived to encrypt information for a single communication session, but the encrypted information is to be retained for a year, the originator usage period of the key when used for encryption is the length of the communication session. The recipient usage period of the key, as used for decryption of the encrypted information, is one year.

13. *Key transport private key*: A key transport private key is used multiple times. The cryptoperiod of the key transport private key may be longer than the cryptoperiod of the associated public key; the cryptoperiod of the private key is the length of time during which any keys encrypted by the associated key transport public key need to be decrypted.

14. *Key transport public key*: The cryptoperiod for the key transport public key is that period of time during which keys may be encrypted by that public key. An appropriate cryptoperiod may be 1-2 years.
15. *Secret key agreement key*: A secret key agreement key is used multiple times. An appropriate cryptoperiod for the key would be 1-2 years.
16. *Static key agreement private key*: A static key agreement private key is used multiple times. The cryptoperiod of the private key and its associated public key would be the same, e.g., for the cryptoperiod of the certificate. An appropriate cryptoperiod for the key would be 1-2 years.
17. *Static key agreement public key*: The cryptoperiod would be the same as the associated static key agreement private key. See the discussion for the *static key agreement private key*. However, a long cryptoperiod for the public key poses no security concern.
18. *Ephemeral key agreement private key*: The cryptoperiod of an ephemeral key agreement private key is the duration of a single key agreement process.
19. *Ephemeral key agreement public key*: The cryptoperiod of an ephemeral key agreement public key is the duration of a single key agreement process. However, a long cryptoperiod for the public key poses no security concern.
20. *Secret authorization key*: A secret authorization key might be used for an extended period of time, depending on the resources that are protected and the role of the entity authorized for access. It is **RECOMMENDED** that cryptoperiods be less than two years.
21. A*uthorization private key*: An authorization private key might be used for an extended period of time, depending on the resources that are protected and the role of the entity authorized for access. The cryptoperiod of the authorization private key and its associated public key **SHOULD** be the same. It is **RECOMMENDED** that cryptoperiods be less than two years.
22. *Authorization public key*: The cryptoperiod of the authorization public key **SHOULD** be the same as the authorization private key: less than two years. However, a long cryptoperiod for the public key poses no security concern.

Other keying material does not have well established cryptoperiods, per se.

- Domain parameters remain in effect until changed.
- An IV is associated with the information that it helps to protect, and is needed until the information and its protection are no longer needed.
- Shared secrets **SHOULD** be destroyed when no longer needed to derive keying material.
- Secret seeds **SHOULD** be destroyed immediately after use.
- It is **RECOMMENDED** that public seeds **NOT** be retained longer than needed for validation (e.g., as one of the elliptic curve domain parameters).
- It is **RECOMMENDED** that nonces **NOT** be retained longer than needed for cryptographic processing.
- Intermediate results **SHOULD** be destroyed immediately after use

3.3 Domain Parameter Validation and Public Key Validation

Domain parameter validation and public key validation techniques are important to many applications of public key cryptography.

Domain parameters are used by some public key algorithms during the generation of key pairs, digital signatures and shared secrets that are subsequently used to derived keying material. Failure to validate these domain parameters could void all intended security for all entities using the domain parameters. The domain parameters used for DSA, ECDSA and the key agreement algorithms **SHOULD** be generated by a trusted party (e.g., a CA), -OR- generated by someone else and validated by a trusted party or the participating entities themselves. The domain parameter validation procedure for DSA and finite field discrete log and MQV key agreement algorithms is provided in ANSI X9.42[17] and [FIPS XXX]. The validation procedure for the elliptic curve discrete log and MQV key agreement algorithms and is provided in [ANSI X9.63] and [FIPS XXX]. The validation procedure for ECDSA is provided in [ANSI X9.62].

Public keys **SHALL** be validated. Failure to validate these keys could result in voiding the intended security, including the security of the operation (i.e., digital signature, key establishment, encryption), leaking some or all information from the owner's private key, and leaking some or all information about a private key that is combined with an invalid public key (as may be done when key agreement or public key encryption is performed). The static public keys used for DSA, ECDSA and the key agreement algorithms could, for example, be validated by a trusted third party (e.g., a CA). Ephemeral public keys used during key establishment could be validated, for example, by the entity that receives the ephemeral key. Public key validation is discussed in [FIPS XXX][18].

[Note: Need to address the validation of public key transport keys for both RSA and EC.]

3.4 Compromise of Keys and other Keying Material

Information protected by cryptographic mechanisms is secure only if the algorithms remain strong and the keys have not been compromised. Key compromise occurs when the protective mechanisms for the key fail (e.g., the confidentiality, integrity or association of the key to its owner fail - see Sections 2.2.3.1 and 2.3.2.2.1), and the key can no longer be trusted to provide the required security. When a key is compromised, all use of the key to apply the protection mechanism on information (e.g., compute a digital signature or encrypt information) **SHOULD** cease, the compromised key **SHOULD** be revoked, and all entities using or relying on that key **SHOULD** be notified (see Section 2.3.3.3). However, continued use of the key to remove or verify the protections (e.g., decrypt or verify a digital signature) may be warranted, depending on the risks of continued use and an organization's Key Management Policy (see Part 2). Limiting the cryptoperiod of the key limits the amount of material that would be compromised (exposed) if the key were compromised. Using different keys for different purposes (e.g., different

---

[17] The DSA domain parameters have the same properties and are created in the same manner as those for ANSI X9.42, so can be validated using the same process.

[18] FIPS XXX refers to ANSI X9.42 and X9.63 for public key validation; the DSA public keys are validated in the same manner as the public keys for ANSI X9.42.

applications as well as different cryptographic mechanisms) as well as limiting the amount of information protected by a single key also achieves this purpose.

The compromise of a key has the following meanings:

- A compromise of the confidentiality of a key means that another entity (an unauthorized entity) may know the key and be able to use that key to perform computations requiring the use of the key. In general, the compromise of a key used to provide confidentiality protection[19] (i.e., via encryption) means that all information encrypted by that key could be known by unauthorized entities. In addition, the encrypted information could contain false information that was originated by an unauthorized entity (an entity that is not authorized to know the key).

  For example, if a secret key (symmetric key) is compromised, the unauthorized entity might use the key to decrypt past or future encrypted information, i.e., the information is no longer confidential between the authorized entities. The unauthorized entity might masquerade as a legitimate entity and send false information, i.e., the authenticity and source of the information would be in question.

  As another example, if a signing private key is compromised, the unauthorized entity might sign messages as if they were originated by the key's real owner (either new messages or messages that are altered from their original contents), i.e., non-repudiation and authenticity of the information is in question.

- A compromise of the integrity of a key means that the key is incorrect - either that the key has been modified (either deliberately or accidentally), or that another key has been substituted; this includes a deletion (non-availability) of the key. The compromise of a key used to provide integrity[20] calls into question the integrity of all information protected by the key. This information could have been provided by, or changed by, an unauthorized entity.

- A compromise of a key's usage or application association means that the key could be used for the wrong purpose (e.g., key establishment instead of digital signatures) or for the wrong application, and could result in the compromise of information protected by the key.

- A compromise of a key's association with the owner or other entity means that the identity of the other entity cannot be assured (i.e., one doesn't know who the other entity really is) or that information cannot be processed correctly (e.g., encrypted or decrypted with the correct key).

- A compromise of a key's association with other information means that there is no association at all, or the association is with the wrong "information". This could cause the

[19] As opposed to the confidentiality of a key that could, for example, be used as a signing private key.

[20] As opposed to the integrity of a key that could, for example, be used for encryption.

cryptographic services to fail, information to be lost, or the security of the information to be compromised.

Certain protective measures may be taken in order to minimize the likelihood of a key compromise. The following procedures are usually involved:

- Limiting the amount of time a secret or private key is in plaintext form.
- Restricting plaintext secret and private keys to approved key containers. This includes key generators, key transport devices, key loaders, cryptographic modules, and key storage devices.
- Preventing humans from viewing plaintext secret and private keys.
- Using integrity checks to assure that the integrity of a key or its association with other data has not been compromised. For example, keys may be wrapped (i.e., encrypted) in such a manner that unauthorized modifications to the wrapping or to the associations will be detected.
- Employing key confirmation and receipt to help ensure that the proper key was, in fact, established.
- Establishing an accountability system that keeps track of each access to secret and private keys in plaintext form.
- Providing a cryptographic integrity check on the key (e.g., a MAC or a digital signature).
- Destroying keys as soon as they are no longer needed.

The worst form of key compromise is one that is not detected. Nevertheless, even in this case, certain protective measures can be taken. Key management systems (KMS) **SHOULD** be designed to mitigate the negative effects of a key compromise. A KMS **SHOULD** be designed so that the compromise of a single key compromises as few other keys as possible. For example, a single cryptographic key could be used to protect the data of only a single user or a limited number of users, rather than a large number of users. Often, systems have alternative methods to authenticate communicating entities that do not rely solely on the possession of keys. The object is to avoid building a system with catastrophic weaknesses.

A compromise recovery plan is essential for restoring cryptographic security services in the event of a key compromise. A compromise recovery plan **SHOULD** be documented and easily accessible. The plan may be included in the Key Management Practices Statement (see Part 2). If not, the Key Management Practices Statement **SHOULD** reference the compromise recovery plan.

Although compromise recovery is primarily a local action, the repercussions of a compromised key are shared by the entire community that uses the system or equipment. Therefore, compromise recovery procedures must include the community at large.

The compromise recovery plan **SHOULD** contain:

- The identification of the personnel to notify,
- The identification of the personnel to perform the recovery actions,
- The rekey or key replacement method, and

- Any other recovery procedures.

Other compromise recovery procedures may include:

- Physical inspection of the equipment,
- Identification of all information that may be compromised as a result of the incident,
- Identification of all signatures that may be invalid due to the compromise of a signing key,
- [List may need to be augmented]

[Note: Discussion could be included on the effects of a compromise on each type of keying material.]

3.5 Accountability

Accountability involves the identification of those who have access to, or control of, cryptographic keys throughout their lifecycles. Accountability can be an effective tool to help prevent key compromises and to reduce the impact of compromises once they are detected. As a minimum, the key management system **SHOULD** account for all individuals who are able to view plaintext cryptographic keys. In addition, more sophisticated key management systems may account for all individuals authorized access to, or control of, any cryptographic keys, whether in plaintext or ciphertext form. For example, a sophisticated accountability system might be able to determine each individual who had control of any given key over its entire life span. This would include the person in charge of generating the key, the person who used the key to cryptographically protect data, and the person who was responsible for destroying the key when it was no longer needed. Even though these individuals never actually saw the key in plaintext form, they are held accountable for the actions that they performed on or with the key.

Accountability provides three significant advantages:

- It aids in the detection of where in the lifecycle the compromise could have occurred and what individuals could have been involved,
- It tends to protect against compromise because individuals with access to the key know that their participation in the key lifecycle is known, and
- It is very useful in recovering from the detected compromise of a key to know where it was used and what data, or other keys, were protected by the compromised key.

Certain principles have been found to be useful in enforcing the accountability of cryptographic keys. These principles might not apply to all systems or all types of keys. Some of the principles apply to longer-term keys that are controlled by humans. The principles include:

- Uniquely identifying keys,
- Identifying the key user,
- Identifying dates and times of key use along with the data that is protected, and
- Identifying other keys that are protected by a secret or private key.

3.6 Audit

Two types of audit **SHOULD** be performed on key management systems:

- The security plan and the procedures that are developed to support the plan **SHOULD** be periodically audited to ensure that they continue to support the Key Management Policy (see Part 2).
- The protective mechanisms employed **SHOULD** be periodically reassessed with respect to the level of security that they provide and are expected to provide in the future, and that they correctly and effectively support the appropriate policies. New technology developments and attacks **SHOULD** be taken into consideration.

On a more frequent basis, the actions of the humans that use, operate and maintain the system **SHOULD** be reviewed to verify that the humans continue to follow established security procedures. Strong cryptographic systems can be compromised by lax and inappropriate human actions.

3.7 Key Management System Survivability

[We may move this material to the Continuity of Operations Key Recovery Section, 2.3.2.3.3.]

[Need to describe data recovery; key recovery is one method of obtaining data recovery.]

3.7.1 Back-up Keys

OMB Guidance to Federal Agencies on Data Availability and Encryption [This should be a reference rather than including the title?] noted that encryption is an important tool for protecting the confidentiality of disclosure-sensitive information that is entrusted to an agency's care, but that the encryption of agency data also presents risks to the availability of information needed for mission performance. Agencies are reminded of the need to protect the continuity of their information technology operations and agency services when implementing encryption. The guidance specifically notes that, without access to the cryptographic keys that are needed to decrypt information, organizations risk the loss of their access to that information. Consequently, it is prudent to retain back-up copies of the keys necessary to decrypt stored enciphered information, including master keys, key encrypting keys, and the related keying material necessary to decrypt encrypted information until there is no longer any requirement for access to the underlying plaintext information (see Tables 4-9).

As the tables show, there are other keys for the operations of some organizations that may require the retention of back-up copies (e.g. signature verification keys and authorization keys). Back-up copies of keying material must be stored in accordance with the provisions of Section 2.2.3.1 in order to protect the confidentiality of encrypted information and the integrity of source authentication, data integrity, and authorization processes.

3.7.2 Key Recovery

There are a number of issues associated with key recovery. An extensive discussion is provided in Appendix B. Key recovery issues to be addressed include:

- Which keying material, if any, need to be backed up or archived for later recovery?
- Where will backed up or archived keying material be stored?

- Who will be responsible for protecting the backed up or archived keying material?
- What procedures need to be put in place for storing and recovering the keying material?
- Who can request a recovery of the keying material and under what conditions?
- Who will be notified when a key recovery has taken place and under what conditions?
- What audit or accounting functions need to be performed to ensure that the keying material is only provided to authorized entities?

3.7.3 System Redundancy/Contingency Planning/Planning

[This section will address recovering data and replacing failed systems.]

3.7.4 Compromise Recovery

[Text to be written]

3.8 Guidance for Cryptographic Algorithm and Key Size Selection

Cryptographic algorithms that provide the security services identified in Section 2.1 are specified in Federal Information Processing Standards (FIPS). Several of these FIPS-approved algorithms are defined for a number of key sizes. This section provides guidance for the selection of appropriate algorithms and key sizes.

This section emphasizes the importance of acquiring cryptographic systems with appropriate algorithm and key sizes to provide adequate protection for 1) the expected lifetime of the system and 2) any data protected by that system during the expected lifetime of the data.

3.8.1 Equivalent Algorithm Strengths

Cryptographic algorithms provide different “strengths” of security, depending on the algorithm and the key size used. In this discussion, two algorithms are considered to be of equivalent strength for the given key sizes (*X* and *Y*) if the amount of time needed to “break the algorithms” or determine the keys (with the given key sizes) is the same. The strength of an algorithm for a given key size is traditionally described in terms of the amount of time it takes to try all keys for a symmetric algorithm with a key size of "*X"* that has no short cut attacks (i.e., the most efficient attack is to try all possible keys). An algorithm that has a "*Y*" bit key, but whose strength is equivalent to an "*X*" bit key of such a symmetric algorithm is said to provide “*X* bits of security” or to provide "*X*-bits of strength". Note: Other metrics for algorithm equivalence exist, e.g., a cost-based model where the amount of memory is considered in addition to the amount of time needed to apply the attack.

The recommended key size equivalencies discussed in this section are based on assessments made as of the publication of this guideline. Advances in factoring algorithms and quantum computing may affect these equivalencies in the future. In the case of quantum computing, the asymmetric techniques may no longer be secure. Periodic reviews will be performed to determine whether the stated equivalencies need to be revised (e.g., the key sizes need to be increased) or the algorithms are no longer secure.

When selecting a block cipher cryptographic algorithm (e.g., AES or TDES), the block size may also be a factor that should be considered, since the amount of security provided by several of the

modes defined in [MODES] is dependent on the block size[21]. More information on this issue is provided in [MODES].

Table 8 provides equivalence guidelines for the Approved algorithms.

- Column 1 indicates the number of bits of security provided by the algorithms and key sizes in a particular row.
- Column 2 provides the symmetric key algorithms that provide the indicated level of security (at a minimum), where TDES is approved in [FIPS 46-3] and specified in [ANSI X9.52], and AES is specified in [FIPS 197]. The table entry for TDES assumes the use of three distinct keys.
- Column 3 provides the equivalent hash algorithms that are specified in FIPS180-2 for the given level of security.
- Column 4 indicates the size of the parameters associated with the standards that use discrete logs and finite field arithmetic (DSA as defined in [FIPS186-3] for digital signatures, and Diffie-Hellman (DH) and MQV key agreement as defined in [ANSI X9.42] and [FIPS XXX]), where L is the size of the modulus p, and N is the size of q. L is commonly considered to be the key size for the algorithm, although L is actually the key size of the public key, and N is the key size of the private key. [Note: While FIPS 180-3 does not yet exist, the stated sizes are planned for the revision to FIPS 186-2.]
- Column 5 defines the value for k (the size of the modulus n) for the RSA algorithm specified in [ANSI X9.31] and [PKCS #1] and adopted in [FIPS 186-2] for digital signatures, and specified in [ANSI X9.44] and adopted in [FIPS XXX] for key establishment. The value of k is commonly considered to be the key size.
- Column 6 defines the value of f (the size of n, where n is the order of the base point G) for the discrete log algorithms using elliptic curve arithmetic that are specified for digital signatures in [ANSI X9.62] and adopted in [FIPS 186-2], and for key establishment as specified in [ANSIX9.63] and adopted in [FIPS XXX]. The value of f is commonly considered to be the key size.

Table 8: Equivalent strengths.

Bits of security	Symmetric key algs.	Hash algs.	DSA, D-H, MQV	RSA	Elliptic Curves
80		SHA-1	$L = 1024$ $N = 160$	$k = 1024$	$f = 160$

[21] Suppose that the block size is b bits. The collision resistance of a MAC is limited by the size of the tag and collisions become probable after $2^{b/2}$ messages, if the full b bits are used as a tag. When using the Output Feedback mode of encryption, the maximum cycle length of the cipher can be at most 2^b blocks; the average cipher length is less than 2^b blocks. When using the Cipher Block Chaining mode, plaintext information is likely to begin to leak after $2^{b/2}$ blocks have been encrypted with the same key.

112	TDES		$L = 2048$ $N = 224$	$k = 2048$	$f = 224$
128	AES-128	SHA-256	$L = 3072$ $N = 256$	$k = 3072$	$f = 256$
192	AES-192	SHA-384	$L = 7680$ $N = 384$	$k = 7680$	$f = 384$
256	AES-256	SHA-512	$L = 15360$ $N = 512$	$k = 15360$	$f = 512$

3.8.2 Defining Appropriate Algorithm Suites

Many applications require the use of several different cryptographic algorithms. When several algorithms can be used to perform the same service, some algorithms are inherently more efficient because of their design (e.g., AES has been designed to be more efficient than Triple DES).

In many cases, a variety of key sizes may be available for an algorithm. For some of the algorithms (e.g., public key algorithms, such as RSA), the use of larger key sizes than are required may impact operations, e.g., larger keys may take longer to generate or longer to process the data. However, the use of key sizes that are too small may not provide adequate security.

Table 9 provides recommendations that may be used to select an appropriate suite of algorithms and key sizes for Federal Government unclassified applications. A minimum of eighty bits of security **SHOULD** be provided by most applications until 2015. Between 2016 and 2035, a minimum of 112 bits of security **SHOULD** be provided. Thereafter, that at least 128 bits of security **SHOULD** be provided.

- Column 1 indicates the years during which the algorithms specified in subsequent columns are appropriate for use.
- Column 2 identifies appropriate symmetric key algorithms and key sizes: the Triple DES algorithm (TDES) is specified in [FIPS46-3], the AES algorithm is specified in [FIPS 197], and the computation of Message Authentication Codes (MACs) using block ciphers is specified in [MODES]. [Note: This is written in anticipation of the specification of MAC modes in the near future.]
- Column 3 specifies the hash sizes to be used for most hash function applications (e.g., digital signatures). Hash functions are specified in [FIPS 180-2].
- Column 4 specifies the hash function and minimum key size to be used for keyed-hash (HMAC) computations. HMAC is specified in [FIPS 198].
- Column 5 indicates the minimum size of the parameters associated with DSA as defined in FIPS [186-3]. [Note: This is specified in anticipation of the sizes to be specified in FIPS 186-3.]

- Column 6 defines the minimum size of the modulus for the RSA algorithm specified in [ANSI X9.31] and [PKCS #1] and adopted in [FIPS 186-2] for digital signatures, and specified in ANSI X9.44 and adopted in [FIPS XXX] for key establishment. [Need to verify what is actually in X9.44.]
- Column 7 defines the minimum size of the base point for the elliptic curve algorithms specified for digital signatures in [ANSI X9.62] and adopted in [FIPS 186-2], and for key establishment as specified in [ANSI X9.63] and adopted in [FIPS XXX].

Table 9: Recommended algorithms and minimum key sizes.

Years	Symmetric key algs. (Encryption & MAC)	Hash Alg.	HMAC	DSA, D-H, MQV	RSA	Elliptic Curves
Present - 2015 (min. of 80 bits of strength)	TDES AES-128 AES-192 AES-256	SHA-1 SHA-256 SHA-384 SHA-512	SHA-1 (≥80 bit key) SHA-256 (≥128 bit key) SHA-384 (≥192 bit key) SHA-512 (≥256 bit key)	Min.: L = 1024; N =160	Min.: k=1024	Min.: f=160
2016 - 2035 (min. of 112 bits of strength)	TDES AES-128 AES-192 AES-256	SHA-256 SHA-384 SHA-512	SHA-256 (≥128 bit key) SHA-384 (≥192 bit key) SHA-512 (≥256 bit key)	Min.: L = 2048 N = 224	Min.: k=2048	Min.: f=224
2036 and beyond (min. of 128 bits of strength)	AES-128 AES-192 AES-256	SHA-256 SHA-384 SHA-512	SHA-256 (≥128 bit key) SHA-384 (≥192 bit key) SHA-512 (≥256 bit key)	Min.: L = 3072 N = 256	Min.: k=3072	Min.: f=256

The algorithms and key sizes in the table are considered appropriate for the protection of data during the given time periods. Algorithms or key sizes not indicated for a given range of years **SHOULD NOT** be used to protect information during that time period. If the security life of information extends beyond one time period specified in the table into the next time period (the later time period), the algorithms and key sizes specified for the later time **SHOULD** be used. The following examples are provided to clarify the use of the table:

(1) If information is originally protected in 2005, and the maximum expected security life of that data is for only ten years, any of the algorithms or key sizes in the table may be used.

(2) If a CA signature key and all certificates issued under that key will expire in five years, then the signature and hash algorithm used to sign the certificate needs to be secure for at least five years. A certificate issued in 2005 using 1024 bit DSA and SHA-1 would be

acceptable; issuing a certificate in 2015 using the same algorithm, key size and hash function would not provide adequate security for the certificate.

(3) If information is initially protected in 2014 and needs to remain secure for a maximum of ten years (i.e., from 2014 to 2023), SHA-1 or a 1024 bit RSA key would not provide sufficient protection between 2016 and 2023 and, therefore, **SHOULD NOT** be used. The algorithms and key sizes in the "2016-2035" row **SHOULD** be used to provide the cryptographic protection.

(4) If information is cryptographically protected in 2013, and needs to remain secure for 30 years (i.e., from 2013 to 2042), then the algorithms and key sizes in the "2036 and Beyond" row **SHOULD** be used.

Algorithms of different strengths and key sizes may be used together for performance, availability or interoperability reasons, provided that sufficient protection is provided. In general, the weakest algorithm and key size used to provide cryptographic protection determines the strength of the protection. Exceptions to this "rule" require extensive analysis. Determination of the security strength provided for protected information includes an analysis not only of the algorithm(s) and key size(s) used to apply the cryptographic protection(s) to the information, but also any algorithms and key sizes associated with establishing the key(s) used for information protection, including those used by communication protocols.

The following is a list of several algorithm combinations and discussions on the security implications of the combination:

- When a key establishment scheme is used to establish keying material for use with one or more algorithms (e.g., TDES, AES, MAC, HMAC), the strength of the selected combination is equivalent to the weakest algorithm and key size used. For example, if a 1024 bit RSA key is used to establish a 128-bit AES key (as defined in [FIPS XXX]), only 80 bits of security are provided for any information protected by that AES key, since the 1024 bit RSA provides only 80 bits of security. If 128 bits of security are required for the information protected by AES, then either an RSA key size of at least 3072 bits, or another key establishment algorithm and appropriate key size needs to be selected to provide the required protection.

- When a hash function and digital signature algorithm are used in combination to compute a digital signature, the strength of the signature is determined by the weaker of the two algorithms. For example, SHA-256 used with ECDSA using a 160-bit key provides 80 bits of security because a 160-bit ECDSA key provides only 80 bits of security. If 112 bits of security is required, a 224-bit ECDSA key would be appropriate; if 128 bits of security is required, a 256-bit ECDSA key would be appropriate.

- When encryption is used with a MAC or a digital signature to protect information, the protection provided for the information is equivalent to the protection provided by the weaker algorithm. For example, TDES used with a SHA-1 HMAC using an 80-bit key to protect a message, provides 80 bits of security; TDES used with a SHA-256 HMAC using a 128 bit key provides 112 bits of security; TDES used with a TDES MAC provides 112 bits of security. As another example, AES used with a 1024-bit DSA key provides 80 bits of security; AES used with a 2048-bit DSA key provides 112 bits of security.

- [Other combinations to be considered?]

Typically, an organization selects the cryptographic services that are needed for a particular application. Then, based on the security life of the data and the years the system is anticipated to be in use, an algorithm and key size suite is selected that is sufficient to meet these requirements. The organization then establishes a key management system, including approved cryptographic products that provide the services required by the application. Finally, when the current algorithm and key size suite nears its expiration date as determined by the security life of the information to be protected (see Table 9), the organization transitions to a new algorithm and key size suite offering appropriate security.

If it is determined that a certain level of security is required for the protection of certain information, then an algorithm and key size suite needs to be selected that would provide that level of security as a minimum. For example, if 128 bits of security are required for information that is to be communicated and provided with confidentiality, integrity, authentication and non-repudiation protection, the following selection of algorithms and key sizes may be appropriate:

- Confidentiality: Encrypt the information using AES-128. Other AES key sizes would also be appropriate, but perform a bit slower.
- Integrity, authentication and non-repudiation: Suppose that only one cryptographic operation is preferred. Use digital signatures. SHA-256 could be selected for the hash function, although SHA-384 and SHA-512 are also appropriate, but the performance is slower. Select an algorithm for digital signatures from what is available to an application (e.g., DSA or RSA with at least a 3072-bit key, or ECDSA with at least a 256-bit key). If more than one algorithm and key size is available, the selection may be based on algorithm performance, memory requirements, etc. as long as the minimum requirements are met.
- Key establishment: Select a key establishment scheme based on the application and environment (see [FIPS XXX]), the availability of an algorithm in an implementation, and its performance. Select a key size from Tables 8 and 9 for the algorithm that provides at least 128 bits of security. For example, if RSA is available, use an RSA key transport (i.e., key establishment) scheme with a 3072-bit key. However, the key used for key transport **SHOULD** be different from an RSA key used for digital signatures.

Agencies that procure systems **SHOULD** consider the potential operational lifetime of the system. The agencies **SHOULD** either select algorithms that are expected to be secure during the entire system lifetime, or **SHOULD** ensure that the algorithms and key sizes can readily be updated.

3.8.3 Transitioning to New Algorithms and Key Sizes

There are many legacy applications currently available that use algorithms and key sizes not specified in Table 8. When the algorithm or key size is determined to no longer provide the desired protection for information (e.g., the algorithm has been "broken"), any information "protected" by the algorithm or key size is considered to be "exposed" (e.g., no longer confidential, or the integrity cannot be assured). It **SHOULD** be assumed that encrypted information could have been collected and retained by unauthorized entities (adversaries). At some time the unauthorized entity may attempt to decrypt the information. Even when an algorithm and key size suite used by a system is replaced (e.g., by a different algorithm or key

size), the information protected by the previous algorithm and key size suite is vulnerable. Thus, when using Table 9 to select the appropriate key size for an encryption algorithm, it is very important to take the expected security life of the data into consideration.

Transition from one algorithm or key size to another is difficult because the information that was considered to be protected by the previous algorithm or key size can no longer be considered as protected. Therefore, when initiating cryptographic protections for information, the strongest algorithm and key size that is appropriate for providing the protection **SHOULD** be used in order to delay transitions. However, it should be noted that selecting some algorithms, or key sizes that are unnecessarily large may have adverse affects (e.g., performance may be unacceptably slow).

3.9 Key Establishment Schemes

Schemes for the establishment of keying material are provided in [FIPS XXX]. Methods have been provided for both key agreement and key transport. Key agreement schemes use asymmetric (public key) techniques. Key transport schemes use either symmetric (secret key) or asymmetric (public key) techniques. Key transport schemes using symmetric techniques are commonly known as key wrapping algorithms.

[Note: discussions will be included on using the schemes in FIPS XXX.]

[Discuss the constraints associated with the use of the key wrapping algorithm, e.g., don't encrypt a 256 bit key with a 128 bit key.]

[Discuss padding?]

[Provide guidance on the use of key confirmation?]

[Provide guidance on how to vary security, e.g., by increasing key sizes. Could be addressed in the section on algorithm and key size selection.]

[Provide a warning in response to Don Johnson's comment: Given all the security requirements stated in the schemes document, some mention of the potential for side-effect attacks seems warranted, as this can unravel a secret/private key by monitoring energy, time, radio waves, etc.]

APPENDIX A: Cryptographic and Non-cryptographic Integrity and Authentication Mechanisms

Integrity and authentication services are particularly important in protocols that include key management. When integrity or authentication services are discussed in this guideline, they are afforded by “strong” cryptographic integrity or authentication mechanisms. Secure communications and key management are typically provided using a communications protocol that offers certain services, such as integrity or a "reliable" transport service. However, the integrity or reliable transport services of communications protocols are not necessarily adequate for cryptographic applications, particularly for key management, and there might be confusion about the meaning of terms such as “integrity”.

All communications channels have some noise (i.e., unintentional errors inserted by the transmission media), and other factors, such as network congestion, can cause packets to be lost. Therefore, integrity and reliable transport services for communications protocols are designed to function over a channel with certain worst-case noise characteristics. Transmission bit errors are typically detected using 1) a non-cryptographic checksum[22] to detect transmission errors in a packet, and 2) a packet counter that is used to detect lost packets. A receiving entity that detects damaged packets (i.e., packets in which errors have been detected) or lost packets may request the sender to retransmit them. The non-cryptographic checksums are generally effective at detecting random noise. For example, the common CRC-32 checksum algorithm used in LAN applications detects all error bursts with a span of less than 32 bits, and detects longer random bursts with a 2^{-32} failure probability. However, the non-cryptographic CRC-32 checksum does not detect the swapping of 32-bit message words, and specific errors in particular message bits cause predictable changes in the CRC-32 checksum. The sophisticated attacker can take advantage of this to create altered messages that pass the CRC-32 integrity checks, even, in some cases, when the message is encrypted.

Forward error correcting codes are a subset of non-cryptographic checksums that can be used to correct a limited number of errors without transmission. These codes may be used, depending on the application and noise properties of the channel.

Cryptographic integrity mechanisms, on the other hand, protect against an active, intelligent attacker who might attempt to disguise his attack as noise. The bits altered by the attacker are not random; they are targeted at system properties and vulnerabilities. Cryptographic integrity mechanisms are effective in detecting random noise events, but they also detect the more systematic deliberate attacks. Cryptographic hash algorithms, such as SHA-1, are designed to make every bit of the hash value a complex, nonlinear function of every bit of the message text, and to make it impractical to find two messages that hash to the same value. On average, it is necessary to perform 2^{80} SHA-1 hash operations to find two messages that hash to the same value, and it is much harder than that to find another message whose SHA-1 hash is the same

[22] Checksum: an algorithm that uses the bits in the transmission to create a checksum value. The checksum value is normally sent in the transmission. The receiver recomputes the checksum value using the bits in the received transmission, and compares the received checksum value with the computed value to determine whether or not the transmission was correctly received. A non-cryptographic checksum algorithm uses a well-known algorithm without secret information (i.e., a cryptographic key).

value as the hash of any given message. Cryptographic message authentication algorithms (MAC) employ hashes or symmetric encryption algorithms and a key to authenticate the source of a message, as well as protect its integrity (i.e., detect errors). Digital signatures use public key algorithms and hash algorithms to provide both authentication and integrity services. Compared to non-cryptographic integrity or authentication mechanisms, these cryptographic services are usually computationally quite expensive; this seems to be unavoidable, since cryptographic protections must resist not just random errors, but deliberate attacks by knowledgeable adversaries with substantial resources.

Cryptographic and non-cryptographic integrity mechanisms may be used together. For example, Consider the TLS protocol (see Part 3). In TLS, a client and a server can authenticate each other, establish a shared "master key" and transfer encrypted payload data. Every step in the entire TLS protocol run is protected by strong cryptographic integrity and authentication mechanisms, and the payload is usually encrypted. Like most cryptographic protocols, TLS will detect any attack or noise event that alters any part of the protocol run. However, TLS has no error recovery protocol. If an error is detected, the protocol run is simply terminated. Starting a new TLS protocol run is quite expensive. Therefore, TLS requires a "reliable" transport service, typically the Internet Transport Control Protocol (TCP), to handle and recover from ordinary network transmission errors. TLS will detect errors caused by an attack or noise event, but has no mechanism to recover from them. TCP will generally detect such errors on a packet-by-packet basis and recover from them by retransmission of individual packets, before delivering the data to TLS. Both TLS and TCP have integrity mechanisms, but a sophisticated attacker could easily fool the weaker non-cryptographic checksums of TCP. However, because of the cryptographic integrity mechanism provided in TLS, the attack is thwarted.

There are some interactions between cryptography and non-cryptographic integrity or error-correction mechanisms that users and protocol designers must take into account. For example, many encryption modes expand ciphertext errors: a single bit error in the ciphertext can change an entire block or more of the resulting plaintext. If forward error correction is applied before encryption, and errors are inserted in the ciphertext during transmission, the error expansion during the decryption might "overwhelm" the error correction mechanism, making the errors uncorrectable. Therefore, it is preferable to apply the forward error correction mechanism after the encryption process. This will allow the correction of errors by the receiving entity’s system before the ciphertext is decrypted, resulting in “correct” plaintext.

Interactions between cryptographic and non-cryptographic mechanisms can also result in security vulnerabilities. One classic way this occurs is with protocols that use stream ciphers[23] with non-cryptographic checksums (e.g. CRC-32) and that acknowledge good packets. An attacker can copy the encrypted packet, selectively modify individual ciphertext bits, selectively

[23] An algorithm that encrypts and decrypts one element (e.g., bit or byte) at a time. There are no FIPS algorithms specifically designated as stream ciphers. However, some of the cryptographic modes defined in [SP 800-38] can be used with a symmetric block cipher algorithm, such as AES, to perform the function of a stream cipher.

change bits in the CRC, and then send the packet. Using the protocol's acknowledgement mechanism, the attacker can determine when the CRC is correct, and therefore, determine when the modified packet is acceptable to the protocol. At least one widely used wireless encryption protocol has been broken with such an attack.

APPENDIX B: Key Recovery

Federal agencies have a responsibility to protect the information contained in, processed by and transmitted between their Information Technology systems. Cryptographic techniques are often used as part of this process. These techniques are used to provide confidentiality, assurance of integrity, non-repudiation or access control. Policies **SHOULD** be established to address the protection and continued accessibility of cryptographically protected information, and procedures **SHOULD** be in place to ensure that the information remains viable during its lifetime. Since cryptographic keying material was used to protect the information, this same keying material may need to be available to remove (e.g., decrypt) or verify (e.g., verify the signature) those protections.

In many cases, the keying material used for cryptographic processes might not be readily available. This might be the case for a number of reasons, including:

- the cryptoperiod of the key has expired, and the keying material is no longer in operational storage,
- the keying material has been corrupted (e.g., the system has crashed or a virus has modified the saved keying material in operational storage), or
- the owner of the keying material is not available, and the owner's organization needs to obtain the protected information.

In order to have this keying material available when required, the keying material needs to be saved somewhere or to be reconstructable (i.e., re-derivable) from other available keying material. The process of re-acquiring the keying material is called key recovery. Key recovery is often used as one method of information recovery when the plaintext information needs to be recovered from encrypted information. However, keying material or other related information may need to be recovered for other reasons, such as the corruption of keying material in normal operational storage (see Section 2.3.2.1), for the verification of digital signatures for archived documents, or checking the integrity of archived information. Key recovery may also be appropriate for situations in which it is easier or faster to recover the keying material than it is to generate and distribute new keying material. Key recovery is motivated by a need to recover or ascertain the validity of cryptographically protected information (e.g., the information that has been encrypted or authenticated) on behalf of an organization or individual.

However, there are applications that may not need to save the keying material for an extended time because of other procedures to recover an operational capability when the keying material or the information protected by the keying material becomes inaccessible. Applications of this type could include telecommunications where the transmitted information could be resent, or applications that could quickly derive, or acquire and distribute new keying material.

It is the responsibility of an organization to determine whether or not the recovery of keying material is required for their application. The decision as to whether key recovery is required **SHOULD** be made on a case by case basis, and this decision **SHOULD** be reflected in the Key Management Policy and the Key Management Practices Statement (see Part 2). If the decision is made to provide key recovery, the appropriate method of key recovery **SHOULD** be selected, based on the type of keying material to be recovered and the capabilities of the organization, and a suitable key recovery methodology **SHOULD** be designed and implemented.

If the decision is made to provide key recovery for a key, all information associated with that key must also be recoverable (see Table 1 in Section 2.2.3.1.1).

[Do we need to discuss and relate backup storage, archive storage, local and remote facilities, and own organization vs. third party key recovery "agent" here?]

B.1 Recovery from Stored Keying Material

The primary purpose of backing up or archiving keying material is to be able to recover that material when it is not otherwise available in normal operational storage to decrypt or check the information protected by the keying material. For example, encrypted information cannot be transformed into plaintext information if the decryption key is lost or modified; the integrity of data cannot be determined if the key used to verify the integrity of that data is not available. The key recovery process acquires the keying material from backup or archive storage, and places it in normal operational storage, either in the device or module, or in immediately accessible storage (see Section 2.3.2.1).

B.2 Recovery by Reconstruction (Rederivation) of Keying Material

Some keying material may be recovered by reconstructing or rederiving the keying material from other available keying material, the “base” keying material (e.g., a master key for a key derivation method). The base keying material **SHOULD** be available in either normal operational storage (see Section 2.3.2.1), backup storage (see Section 2.3.2.2.1) or archive storage (see Section 2.3.3.1).

B.3 Conditions Under Which Keying Material Needs to be Recoverable

The decision as to whether to backup or archive keying material for possible key recovery **SHOULD** be made on a case by case basis. The decision **SHOULD** be based on:

- the type of key (e.g., signing private key, long-term data encryption key),
- the application in which the key will be used (e.g., interactive communications, file storage),
- whether the key is "owned" by the local entity (e.g., a private key) or by another entity (e.g., the other entity's public key) or is shared (e.g., a data encyption key shared by two entities),
- the role of the entity in a communication (e.g., sender of receiver),
- the algorithm or computation in which the key will be used (e.g., does the entity have the necessary to information to perform a given computation if the key were to be recovered)[24], and
- the value of the information protected by the keying material, and the consequences of the loss of the keying material.

[24] This could be the case when performing a key establishment process for some key establishment schemes (see FIPS XXX).

The factors involved in a decision for or against key recovery **SHOULD** be carefully assessed. The trade-offs are concerned with continuity of operations versus the risk of possibly exposing the keying material and the information it protects if control of the keying material is lost. If it is determined that a key needs to be recoverable, and the key is still active (i.e., the cryptoperiod of the key has not expired), then the key **SHOULD** be replaced as soon as possible in order to limit its exposure (see Section 2.3.2.3).

The following subsections provide discussions to assist an organization in determining whether or not key recovery is required. Although the following discussions address only the recoverability of keys, any related information **SHOULD** also be recoverable.

[The following subsections are an attempt to analyze the need for backup, archive and key recovery for the various key types and their uses. Feedback is welcome on its reasonableness and completeness. If appropriate, these discussions may be "collapsed" into shorter discussions at a later time.]

B.3.1 Signature Key Pair

The private key of a signature key pair (the signing private key) is used by the owner of the key pair to apply digital signatures to information. The associated public key (the verification public key) is used by relying entities to verify the digital signature.

B.3.1.1 Signature Verification Public Key

It is appropriate to backup or archive a signature verification public key for as long as required to verify the information signed by the associated signing private key. In the case of a public key that has been certified (e.g., by a Certificate Authority), saving the public key certificate would be an appropriate form of storing the public key; backup or archive storage may be provided by the infrastructure (e.g., by a certificate repository). The public key **SHOULD** be stored in backup storage until the end of the signing private key's cryptoperiod, and **SHOULD** be stored in archive storage as long as required. Storage for a longer period poses no security threat.

B.3.1.2 Signing Private Key

Key backup is not usually desirable for the private key of a signing key pair, since the non-repudiability of the signature comes into question. However, exceptions may exist. For example, replacing the signing private key and having its associated verification public key distributed (in accordance with Section 2.3.1.5.1) in a timely manner may not be possible under some circumstances. This may be the case, for example, for the signing private key of a CA. If a signing private key is backed up, the signing private key **SHOULD** be recovered using a highly secure method. Depending on circumstances, the key **SHOULD** be recovered for immediate use only, and then **SHOULD** be replaced as soon after the recovery process as possible. Instead of backing up the signing private key, a second signing private key and associated public key could be generated, and the public key distributed in accordance with Section 2.3.1.5.1 for use if the primary signing private key becomes unavailable. If backup is considered for the signing private key, an assessment **SHOULD** be made as to its importance and the time needed to recover the key, as opposed to the time needed to generate a key pair, and certify and distribute a new verification public key.

As indicated in Table 4 in Section 2.3.2.2.1, a signing private key may be reside in backup storage during the key's cryptoperiod; however, a signing private key **SHOULD NOT** be archived.

B.3.2 Secret Authentication Key

A secret authentication key is used to provide assurance of the integrity and source of information. A secret authentication key can be used:

(1) by an originator to create a message authentication code (MAC) that can be verified at a later time to determine the authenticity or integrity of the authenticated information; the authenticated information and its MAC could then be stored for later retrieval or transmitted to another entity,

(2) by an entity that retrieves the authenticated information and the MAC from storage to determine the integrity of the stored information (Note: This is not a communication application),

(3) immediately upon receipt by a receiving entity to determine the integrity of transmitted information and the source of that information (the received MAC and the associated authenticated information may or may not be subsequently stored), or

(4) by a receiving and retrieving entity to determine the integrity and source of information that has been received and subsequently stored using the same MAC (and the same authentication key); checking the MAC is not performed prior to storage.

In case 1, the authentication key **NEED NOT** be backed up if the originator can establish a new authentication key prior to computing the MAC, making the key available to any entity that would need to subsequently verify the information that is authenticated using this new key. If a new authentication key cannot be obtained in a timely manner, then the authentication key **SHOULD** be backed up or archived.

In case 2, the secret authentication key **SHOULD** be backed up or archived for as long as the integrity of the information needs to be determined.

In case 3, the key **NEED NOT** be backed up if the authentication key can be resent to the recipient. In this case, establishing and distributing a new secret authentication key rather than reusing the “lost” key is also acceptable; a new MAC would need to be computed on the information using the new authentication key. Otherwise, the secret authentication key **SHOULD** be backed up. Archiving the authentication key is not appropriate if the MAC and the authenticated information are not subsequently stored, since the use of the key for both applying and checking the MAC would be discontinued at the end of the key's cryptoperiod. If the MAC and the authenticated information are subsequently stored, then the secret authentication key **SHOULD** be backed up or archived for as long as the integrity and source of the information needs to be determined.

In case 4, the secret authentication key **SHOULD** be backed up or archived for as long as the integrity and source of the information needs to be determined.

The secret authentication key may be stored in backup storage for the cryptoperiod of the key, and in archive storage until no longer required. If the authentication key is recovered by reconstruction, the “base” key (e.g., the master key for a key derivation method) may be stored in normal operational storage or backup storage for the cryptoperiod of the key, and in archive storage until no longer required.

B.3.3 Authentication Key Pair

An authentication public key is used by a receiving entity to obtain assurance of the identity of the entity that originated information and the integrity of the information. The associated authentication private key is used by the originating entity to provide this assurance to a receiving entity by computing a digital signature on the information. This key pair may not provide non-repudiation.

B.3.3.1 Authentication Public Key

It is appropriate to store an authentication public key in either backup or archive storage for as long as required to verify the authenticity of the data that was authenticated by the associated authentication private key. In the case of a public key that has been certified (e.g., by a Certificate Authority), saving the public key certificate would be an appropriate form of storing the public key; backup or archive storage may be provided by the infrastructure (e.g., by a certificate repository). The public key may be stored in backup storage until the end of the private key's cryptoperiod, and may be stored in archive storage as long as required. Storage for a longer period poses no security threat.

B.3.3.2 Authentication Private Key

When the private key is used only for the authentication of transmitted data, whether or not the authenticated data is subsequently stored, the authentication private key **NEED NOT** be backed up if a new key pair can be generated and the distributed in accordance with Section 2.3.1.5.1 in a timely manner. However, if a new key pair cannot be generated, the private key **SHOULD** be stored in backup storage during the cryptoperiod of the private key. The private key **SHOULD NOT** be stored in archive storage.

When the private key is used to protect stored information only, the authentication private key **SHOULD NOT** be backed up if a new key pair can be generated. However, if a new key pair cannot be generated, the private key **SHOULD** be stored in backup storage during the cryptoperiod of the private key. The private key **SHOULD NOT** be stored in archive storage.

B.3.4 Data Encryption Key

A data encryption key (i.e., either a long-term or short-term data encryption key) is used to protect the confidentiality of stored or transmitted information or both.

Information that is stored needs to be readily available as long as the information is encrypted (i.e., for as long as the information may need to be recovered); this includes information that is both transmitted and stored using the same key. The decryption key (which was also used to encrypt the information) needs to be available during the period of time that the information may need to be recovered. Therefore, the key **SHOULD** be backed up or archived during this period.

In order to allow key recovery, the data encryption key **SHOULD** be stored in backup storage during the cryptoperiod of the key, and stored in archive storage after the end of the key's cryptoperiod.

A data encryption key used for transmission only is used by an originating entity to encrypt information, and by the receiving entity to decrypt the information immediately upon receipt. If the data encryption key is lost or corrupted, and a new data encryption key can be easily obtained by the originating and receiving entities, then the key need not be backed up. However, if the key cannot be easily replaced by a new key, then the key **SHOULD** be backed up if the information

to be exchanged is of sufficient importance. The data encryption key **SHOULD NOT** be archived when used for transmission only.

B.3.5 Random Number Generation Key

A key used for random (i.e., pseudorandom) number generation **SHOULD NOT** be backed up or archived. If this key is lost or modified, it **SHOULD** be replaced with a new key.

B.3.6 Key Wrapping Key

A key wrapping key is used to wrap (i.e., encrypt) keying material that is to be protected. The keying material is then transmitted or stored or both. The long-term and short-term key process is performed and the potential number of "sets of keying material" that are protected. Both types may be used during transmission, storage or both.

B.3.6.1 Long-term Key Wrapping Key

A long-term key wrapping key is typically used to protect multiple sets of keying material.

If a long-term key encrypting key is used only to transmit keying material, and the key wrapping key becomes unavailable (e.g., is lost or corrupted), it may be possible to either resend the key wrapping key, or to establish a new key wrapping key and use it to resend the keying material. If this is possible with in a reasonable timeframe, backup of the key wrapping key is not necessary. If the key wrapping key cannot be resent or a new key wrapping key cannot be readily obtained, backup of the key wrapping key **SHOULD** be considered. The archive of a long-term key wrapping key that is only used to transmit keying material is not necessary.

If a long-term key wrapping key is only used to protect keying material in storage, then the key wrapping key **SHOULD** be backed up or archived for as long as the keying material may need to be accessed.

If a long-term key wrapping key is used for the protection of keying material during both transmission and subsequent storage, then the key wrapping key **SHOULD** be backed up or archived for as long as the keying material may need to be accessed.

B.3.6.2 Short-term Key Wrapping Key

A short-term symmetric key wrapping key is used to wrap a single "set" of keying material during a single key transport process (e.g., during a communication session) or a single storage process. A key wrapping key might, for example, be created using a key agreement process and subsequently used to transport the keying material during communication. A new key wrapping key might be created each time keying material needs to be stored.

If the key wrapping key is used during transmission only (i.e., not also used to protect keying material during storage), and the key wrapping key is lost or corrupted, then the key wrapping key **SHOULD** be resent or replaced. The keying material **SHOULD** be wrapped in the active key. In this case, the key wrapping key **SHOULD NOT** be backed up or archived.

If the key wrapping key is subsequently used to wrap the keying material during storage, then the key wrapping key **SHOULD** be backed up or archived for as long as the keying material may need to be accessed.

B.3.7 Master Key used for Key Derivation and the Derived Key

A master key is normally used to derive one or more other keys. It **SHALL NOT** be used for any other purpose.

B.3.7.1 Master Key Used for Key Derivation

The determination as to whether or not a master key needs to be backed up or archived depends on a number of factors:

- How easy is it to establish a new master key? If the master key is distributed manually (e.g., in smart cards or in hard copy by receipted mail), the master key **SHOULD** be backed up or archived. If a new master key can be easily and quickly established using electronic key establishment protocols, then the backup or archiving of the master key may not be desirable, depending on the application.
- Are the derived keys recoverable without the use of the master key? If the derived keys do not need to be backed up or archived (e.g., because of their use) or recovery of the derived keys does not depend on reconstruction from the master key (e.g., the derived keys are stored in an encrypted form), then the backup or archiving of the master key may not be desirable. If the derived keys need to be backed up or archived, and the method of key recovery requires reconstruction of the derived key from the master key, then the master key **SHOULD** be backed up or archived.

B.3.7.2 Key Derived from a Master Key

The use of the derived key **SHOULD** be used to determine whether or not the derived key needs to be backed up or archived (see the discussions for the appropriate key in this appendix).

B.3.8 Key Transport Key Pair

A key transport key pair may be used to transport keying material from an originating entity entity to a receiving entity during communications, or to protect keying material while in storage. The originating entity in a communication (or the entity initiating the storage of the keying material) uses the public key to encrypt the keying material; the receiving entity (or the entity retrieving the stored keying material) uses the private key to decrypt the encrypted keying material.

B.3.8.1 Key Transport Private Key

If a key transport key pair is only used during communications, then the private key does not need to be backed up if a replacement key pair can be generated and distributed in a timely fashion. Otherwise, the private key **SHOULD** be backed up. Alternatively, one or more additional key pairs could be made available (i.e., already generated and distributed). Archiving of the private key is inappropriate.

If the transport key pair is used only during storage, then the private key **SHOULD** be backed up or archived for as long as the protected keying material may need to be accessed.

If the transport key pair is used during both communications and storage of keying material, then the private key **SHOULD** be backed up or archived for as long as the protected keying material may need to be accessed.

B.3.8.2 Key Transport Public Key

If the sending entity (the originating entity in a communications) loses the public key or determines that the key has been corrupted, the key can be reacquired from the key pair owner or by obtaining the public key certificate containing the public key (if the public key was certified).

If the entity that applies the cryptographic protection to keying material that is to be stored determines that the public key has been lost or corrupted, the entity may recover in one of the following ways:

- If the public key has been certified and is stored elsewhere within the infrastructure, then the certificate can be requested.
- If some other entity knows the public key (e.g., the other entity is the actual owner of the key pair), the key can be requested from this other entity.
- If the private key is known, then the public key can be recomputed.
- A new key pair can be generated.

B.3.9 Secret Key Agreement Key

Secret key agreement keys are used to establish keying material (e.g., key encrypting keys used for key wrapping, data encryption keys, or MAC keys). Each key agreement key is shared between two or more entities. If these keys are distributed manually (e.g., in a key loading device or by receipted mail), then the secret key agreement key **SHOULD** be backed up. If an electronic means is available for quickly establishing new keys (e.g., a key transport mechanism can be used to establish a new secret key agreement key), then a secret key agreement key **NEED NOT** be backed up. Secret key agreement keys **SHOULD NOT** be archived.

B.3.10 Static Key Agreement Key Pair

A static key agreement key pairs are used to establish shared secret between entities, often in conjunction with ephemeral key pairs (see FIPS XXX). Each entity uses their private key agreement key(s), the other entity's public key agreement key(s) and possibly their own public key agreement key(s) to determine the shared secret. The shared secret is subsequently used to derive shared keying material. Note that in some key agreement schemes, one or more of the entities may not have a static key agreement pair (see FIPS XXX).

B.3.10.1 Static Key Agreement Private Key

If the private key cannot be replaced in a timely manner, then the static key agreement private key **SHOULD** be backed up in order to continue operations. The private key **SHOULD NOT** be archived **UNLESS** needed for reconstruction of the keying material.

B.3.10.2 Static Key Agreement Public Key

If an entity determines that the public key is lost or corrupted, the entity recover in one of the following ways:

- If the public key has been certified and is stored elsewhere within the infrastructure, then the certificate can be requested.
- If some other entity knows the public key (e.g., the other entity is the actual owner of the key pair), the key can be requested from this other entity.

- If the private key is known, then the public key can be recomputed.
- If the entity is the owner of the key pair, a new key pair can be generated and distributed.

If none of these alternatives are possible, then the key agreement public key **SHOULD** be backed up. The public key **SHOULD NOT** be archived unless needed for reconstruction of the keying material.

B.3.11 Ephemeral Key Pairs

Ephemeral key agreement keys are generated and distributed during a single key agreement process (e.g., at the beginning of a communication session) and are not reused. These key pairs are used to establish a shared secret (often in combination with static key pairs); the shared secret is subsequently used to derive shared keying material. Not all key agreement schemes use ephemeral key pairs, and when used, not all entities have an ephemeral key pair (see FIPS XXX).

B.3.11.1 Ephemeral Private Keys

Ephemeral private keys **SHOULD NOT** be backed up or archived. If the ephemeral private key is lost or corrupted, a new key pair **SHOULD** be generated, and the ephemeral public key **SHOULD** be provided to the other participating entity in the key agreement process.

B.3.11.2 Ephemeral Public Keys

Ephemeral public keys may be backed up or archived if they are required for reconstruction of the established keying material, and the other entity's ephemeral private key is not required in the key agreement computation.

B.3.12 Secret Authorization Key

Secret authorization keys are used to provide privileges to an entity (e.g., access to certain information or authorization to perform certain functions). Loss of these keys will deny the privileges (e.g., prohibit access and disallow performance of these functions). If the authorization key is lost or corrupted and can be replaced in a timely fashion, then the authorization key need not be backed up. A secret authorization key **SHOULD NOT** be archived.

B.3.13 Authorization Key Pair

Authorization key pairs are used to provide privileges to an entity. The private key is used to establish the "right" to the privilege; the public key is used to determine that the entity actually has the right to the privilege.

B.3.13.1 Authorization Private Key

Loss of the private key will deny the privileges (e.g., prohibit access and disallow performance of these functions). If the private key is lost or corrupted and can be replaced in a timely fashion, then the private key need not be backed up. Otherwise, the private key **SHOULD** be backed up. The private key **SHOULD NOT** be archived, since the privilege will not be granted after the end of the cryptoperiod unless a new authorization key pair is provided.

B.3.13.2 Authorization Public Key

If the authorization key pair can be replaced in a timely fashion (i.e., regeneration of the key pair and secure distribution of the private key to the entity seeking authorization), then the public key

need not be backed up. Otherwise, the public key **SHOULD** be backed up. The archive of the public key is not appropriate.

B.3.14 Other Keying Material

Like keys, other keying material may need to be backed up or archived, depending on use.

B.3.14.1 Domain Parameters

Domain parameters are used in conjunction with some public key algorithms to generate key pairs (for digital signature generation or verification or for key establishment) or to create and verify digital signatures using those key pairs. The same set of domain parameters is often, but not always) used by a large number of entities.

When an entity (entity A) generates new domain parameters whenever key pairs are generated, these domain parameters are used in subsequent digital signature generation or key establishment processes. The domain parameters need to be provided to other entities that need to verify the digital signatures or with whom keys will be established. If the entity (entity A) determines that its copies of the domain parameters have been lost or corrupted, and if the new domain parameters cannot be securely distributed in a timely fashion, then the domain parameters **SHOULD** be backed up or archived. Another entity (entity B) **SHOULD** backup or archive entity A's domain parameters until no longer required **UNLESS** the domain parameters can be otherwise obtained (e.g., from entity A).

When the same set of domain parameters are used by multiple entities, the domain parameters **SHOULD BE** backed up or archived until no longer required **UNLESS** the domain parameters can be otherwise obtained (e.g., from a trusted source).

B.3.14.2 Initialization Vectors (IVs)

IVs are used by several modes of operation during the encryption or authentication of information using block cipher algorithms. These IVs **SHOULD** be backed up or archived as long as the information protected using those IVs needs to be processed (e.g., decrypted or authenticated).

B.3.14.3 Shared Secrets

Shared secrets are generated by each entity participating in a key agreement process. The shared secret is then used to derive the shared keying material to be used in subsequent cryptographic operations. Shared secrets may be generated during interactive communications (e.g., where both entities are online) or during non-interactive communications (e.g., in store and forward applications).

A shared secret **SHOULD NOT** be backed up or archived UNLESS the shared secret is used to reconstruct keying material during key recovery.

B.3.14.4 Secret Seeds

Secret seeds are used in the generation of pseudorandom random numbers that need to remain secret. These seeds are not shared with other entities and **SHOULD NOT** be reused. Therefore, secret seeds **SHOULD NOT** be backed up or archived.

B.3.14.5 Public Seeds

Public seeds are used in the generation of pseudorandom numbers that may be validated (e.g., in order to validate domain parameters). The public seed **SHOULD** be backed up or archived as long as these numbers need to be validated.

B.3.14.6 Nonces

A nonce is a non-repeating value that is used only once to apply cryptographic protections to information. However, the nonce may need to be available to process the cryptographically protected information (e.g., to decrypt or authenticate); therefore, the nonce **SHOULD** be backed up or archived until no longer needed to process information.

B.3.14.7 Intermediate Results

The intermediate results of a cryptographic operation **SHOULD NOT** be made backed up or archived. The cryptographic operations **SHOULD** be re-initialized or restarted.

B.3.15 Other Cryptographic Information

Other cryptographic information **SHOULD** be backed up or archived if the protected information needs to be recovered.

B.3.15.1 Hash Value

If the hash value cannot be recomputed, then the hash value **SHOULD** be backed up or archived. However, in most cases, the hash value can be recomputed using the protected information, so need not be backed up or archived.

B.3.15.2 Encrypted Data

The encrypted data **SHOULD** be backed up or archived as long as the plaintext form of the data is required but not available.

B.3.15.3 Message Authentication Code (MAC)

A MAC **SHOULD** be backed up or archived for as long as the source and integrity of the information protected by the MAC needs to be verified.

B.3.15.4 Digital Signature

A digital signature **SHOULD** be backed up or archived for as long as the source,integrity and non-repudiability of the information protected by the digital signature needs to be verified.

B.3.15.5 Key Control Information

Key control information is used, for example, to determine the keys and other information to be used to process cryptographically protected information (e.g., decrypt or authenticate), to identify the purpose of a key, or the entities that share the key (see Section 2.2.3.2.3).

Key control information **SHOULD** be backed up or archived for as long as the associated key needs to be available.

B.3.15.6 Random Number

Random numbers are generated by random number generators. The backup or archiving of a random number depends on how it's used. [May need more on this. previously had "Random numbers **SHOULD NOT** be recoverable unless needed for key reconstruction".]

B.3.15.7 Passwords

A password is used to acquire access to privileges to an entity. The loss of a password will deny the privileges. If the password is lost or corrupted and can be replaced in a timely fashion, then the password need not be backed up. A password **SHOULD NOT** be archived.

B.3.15.8 Audit Information

Audit information containing key management events **SHOULD** be backed up and archived.

B.4 Key Recovery Systems

Key recovery is a broad term that may be applied to several different key recovery techniques. Each technique will result in the recovery of a cryptographic key and other information associated with that key (i.e., the keying material). The information required to recover that key may be different for each application or each key recovery technique. The term “Key Recovery Information” (KRI) is used to refer to the aggregate of information needed to recover or verify cryptographically protected information. Information that may be considered as KRI includes the keying material to be recovered or sufficient information to reconstuct the keying material, other associated cryptographic information, the time when the key was created, the identity of the owner of the key (the individual, application or organization who created the key or who own the data protected by that key) and any conditions that must be met by a requestor to be able to recover the keying material.

When an organization determines that key recovery is required for all or part of its keying material, a secure Key Recovery System (KRS) needs to be established in accordance with a well defined Key Recovery Policy (see Appendix A.3.5). The KRS **SHOULD** support the Key Recovery Policy and consists of the techniques and facilities for saving and recovering the keying material, the procedures for administering the system, and the personnel associated with the system.

When key recovery is determined to be necessary, the KRI may be stored either within an organization (in backup or archive storage) or may be stored at a remote site by a trusted entity. There are many acceptable methods for enabling key recovery. A KRS could be established using a safe for keying material storage; a KRS might use a single computer that provides the initial protection of the plaintext information, storage of the associated keying material and recovery of that keying material; a KRS may include a network of computers with a central Key Recovery Center; or a KRS could be designed using other configurations. Since a KRS provides an alternative means for recovering cryptographic keys, a risk assessment should be performed to ensure that the KRS adequately protects the organization’s information and reliably provides the KRI when required. It is the responsibility of the organization that needs to provide key recovery to ensure that the Key Recovery Policy, the key recovery methodology, and the Key Recovery System adequately protect the KRI.

A KRS should:

- Generate or provide sufficient KRI to allow recovery or verification of protected information.
- Ensure the validity of the saved key and the other KRI.
- Ensure that the KRI is stored with persistence and availability that is commensurate with that of the corresponding cryptographically protected data.
- Any cryptographic modules used by the KRS should be compliant with FIPS 140-2.
- The KRS **SHOULD** use FIPS-approved or NIST recommended algorithms, when cryptography is used.
- The strength of any algorithms used to protect KRI should be commensurate with the sensitivity of the information associated with the KRI.
- The KRS **SHOULD** be designed to enforce the Key Recovery Policy (see Section B.3.5).
- The KRS **SHOULD** protect KRI against unauthorized disclosure or destruction. The KRS should verify the source of requests and ensure that only requested and authorized information is provided to the requestor.
- The KRS **SHOULD** protect the KRI from modification.
- The KRS **SHOULD** have the capability of providing an audit trail. The audit trail should not contain the keys that are recovered or any passwords that may be used by the system. The audit trail might include the identification of the event being audited, the time of the event, the identity of the user causing the event, and the success or failure of the event.
- The KRS **SHOULD** limit access to the KRI, the audit trail and authentication data to authorized individuals.
- It should not be possible to modify the audit trail.

B.5 Key Recovery Policy

An organization that determines that key recovery is required for some or all of their keying material should develop a Key Recovery Policy that addresses the protection and continued accessibility of that information. For each system, application and cryptographic technique used, consideration must be given as to whether or not the keying material may need to be saved for later recovery of the keying material to allow subsequent decryption or checking of the information protected by the keying material.

The policy **SHOULD** address (at a minimum):

- The keying material that needs to be saved for a given application. For example, keys and IVs used for the decryption of stored information protected by the keys and IVs may need to be saved. Keys for the authentication of stored or transmitted information may also need to be saved.
- How and where the keying material would be saved. For example, the keying material could be stored in a safe by the individual who initiates the protection of the information (e.g., the encrypted information), or the keying material could be saved automatically

when the protected information is transmitted, received or stored. The keying material could be saved locally or at some remote site.

- Who will be responsible for protecting the KRI. Each individual, organization or sub-organization could be responsible for their own keying material, or an external organization could perform this function.
- Who can request key recovery and under what conditions. For example, the individual who protected the information (i.e., used and stored the KRI) or the organization to which the individual is assigned could recover the keying material. Legal requirements may need to be considered. An organization could request the information when the individual who stored the KRI is not available.
- Under what conditions could the policy be modified and by whom.
- What audit capabilities and procedures would be included in the KRS. The policy should identify the events to be audited. Auditable events might include KRI requests and their associated responses; the startup and shutdown of audit functions; the operations performed to read, modify or destroy the audit data; requests to access user authentication data; and the uses of authentication mechanisms.
- How the KRS would deal with aged keying material or the destruction of the keying material.
- Who would be notified when keying material is recovered and under what conditions. For example, the individual who encrypted data and stored the KRI could be notified when the organization recovers the decryption key because the person is absent, but the individual might not be notified when the organization is monitoring the activities of that individual.
- The procedures that need to be followed when the KRS or some portion of the data within the KRS is compromised.

B.6 Other Questions/Issues?

APPENDIX C: Cryptoperiods for Signing Key Pairs

[Text to be provided.]

APPENDIX X: References

[AC] Applied Cryptography, Schneier, John Wiley & Sons, 1996.

[ANSI-X9.31] Digital Signatures Using reversible Public Key Cryptography for the Financial Services Industry (rDSA), 1998

[ANSI-X9.42] Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using discrete Logarithm Cryptography, 2001

[ANSI-X9.44] Public Key Cryptography for the Financial Services Industry: Key Agreement Using Factoring-Based Cryptography, [Insert Date]

[ANSI-X9.52] Triple Data Encryption Algorithm Modes of Operation, July, 1998

[ANSI-X9.62] Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1998

[ANSI-X9.63] Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography, [Insert Date]

[FIPS46-3] Data Encryption Standard (DES), October 25, 1999

[FIPS140-2] Security Requirements for Cryptographic Modules, May 25, 2001

[FIPS180-2] Secure Hash Standard (SHS), January 2000.

[FIPS186-2] Digital Signature Standard (DSS), June 2000.

[FIPS197] Advanced Encryption Standard (AES), November 2001.

[FIPS198] Keyed-Hash Message Authentication Code (HMAC), [Insert Date]

[FIPS XXX] [Key Establishment Schemes Standard], [Insert date]

[HAC] Handbook of Applied Cryptography, Menezes, van Oorschot and Vanstone, CRC Press, 1996.

[IPKI] Introduction to Public Key Cryptography and the Federal PKI Infrastructure; Kuhn, Hu, Chang and Polk; draft NIST Special Publication, 2001 [available at http://csrc.nist.gov/publications]

[ITL Bulletin] Techniques for System and Data Recovery, NIST ITL Computer Security Bulletin, April 2002

[MODES] Special Publication 800-38, Recommendation for Block Cipher Modes of Operation, December 2001.

[PKCS1] PKCS #1 v2.0: RSA Cryptography Standard, RSA Laboratories, October, 1998.

[SP800-5] A Guide to the Selection of Anti-Virus Tools and Techniques, December 1992

Parking Lot for Issues to be Addressed in a Different Document

1. Point out differences in strength between MACs, hashes, keyed hashes, and non-cryptographic checksums.

2. RNG seed **and key** may be needed for domain parameter generation and validation.